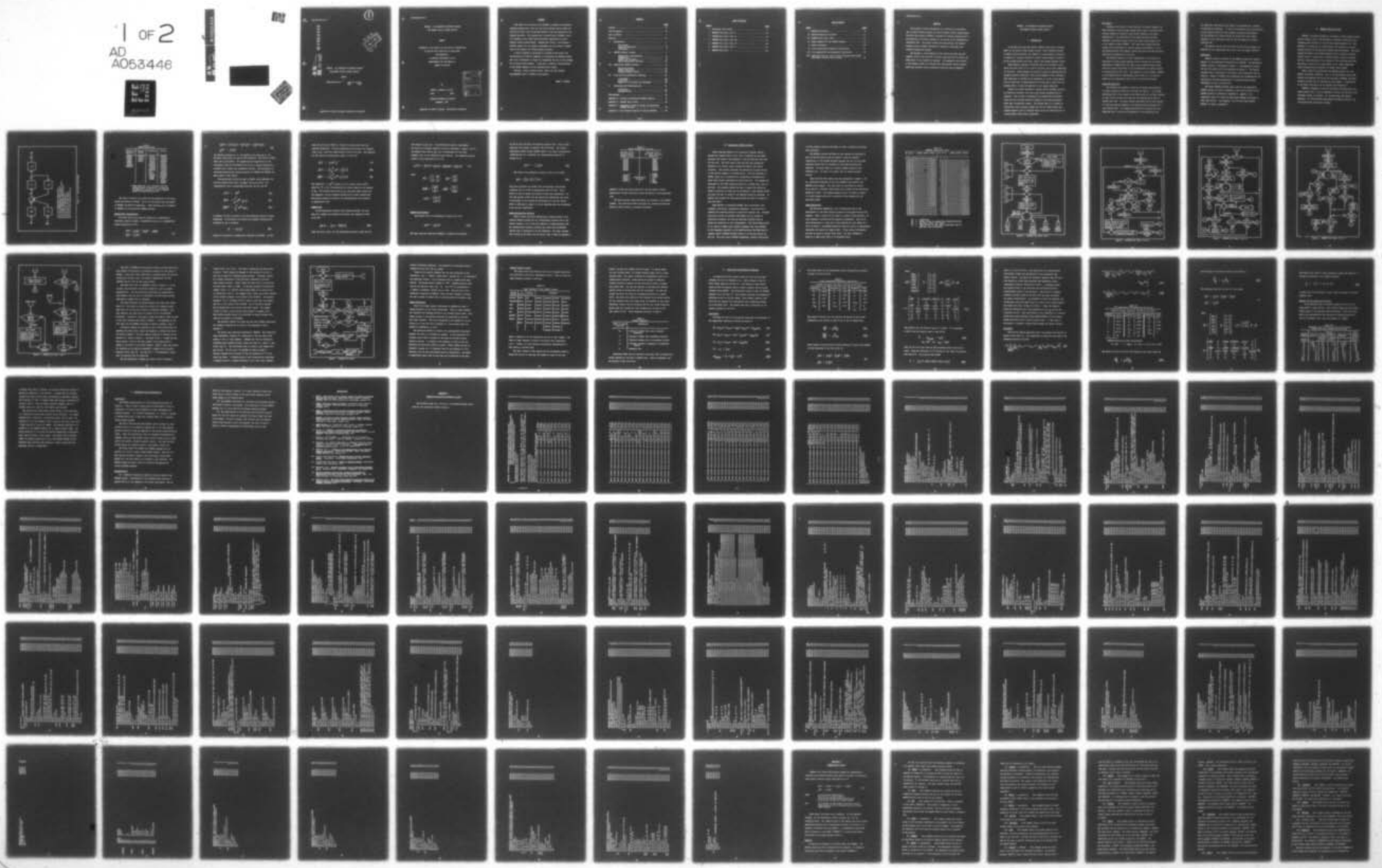
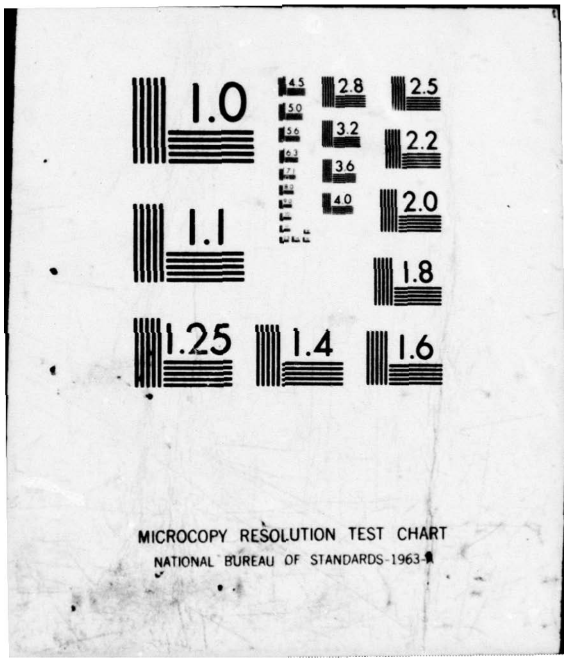
AD-A053 446 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 12/1
INTERAC - AN INTERACTIVE SOFTWARE PACKAGE FOR DIRECT DIGITAL CO--ETC(U)
DEC 77 J A COLGATE
UNCLASSIFIED AFIT/GGC/EE/77D-1 NL

1 OF 2
AD
A053446




1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

(1)

AFIT/GGC/EE/77D-1

INTERAC - AN INTERACTIVE SOFTWARE PACKAGE FOR DIRECT DIGITAL CONTROL DESIGN

THESIS

AFIT/GGC/EE/77D-1

James A. Colgate
Capt USAF

AFIT/GGC/EE/77D-1

INTERAC - AN INTERACTIVE SOFTWARE PACKAGE FOR DIRECT DIGITAL CONTROL DESIGN

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

James A. Colgate, B.S.E.E.
Capt USAF

Graduate Guidance and Control
December 1977

Preface

This report is the result of my attempt to develop an interactive computer program that a user can use to get exactly what he needs (no more and no less) with the minimum amount of work and expertise in the program operation. The program that I started with is FORTRAC, which is a package of the latest state-of-the-art algorithms for direct digital control system design. Through this effort, I have gained a healthy respect for the computer programmers who can produce a simple easy to use program for solving complex problems.

I wish to express my gratitude to Professor Brian Porter from the University of Salford, England, for developing the FORTRAC package and to Dr. Constantine H. Houpis for suggesting the use of this package for a control system design. I also wish to thank Mr. Duane Robertus of the Flight Dynamics Lab for sponsoring this thesis.

Finally, I wish to thank my wife, Julie, for her constant encouragement while I worked on this project.

James A. Colgate

Contents

List of Figures

List of Tables

Abstract

The purpose of this investigation is to develop an interactive, user oriented software package for direct digital control system design. The batch mode program, FORTRAC, developed by Professor Brian Porter of the University of Salford, England, is the source of the computational algorithms used. This report details the input,output, and program sequence control software developed to produce an efficient, user oriented interactive package.

The package is very forgiving of user errors and gives the user complete control over what data to input, what data will be output, and which parts of the program are executed. The package is quite useful in the design process for discrete-time time-optimal control systems, where many possible control parameter variations must be examined.

INTERAC - AN INTERACTIVE SOFTWARE PACKAGE FOR DIRECT DIGITAL CONTROL DESIGN

I. Introduction

In the past few years the digital computer has become a valuable asset in the control system design process. With increased interest in modern control theories, the computer is fast becoming a necessity. A portion of a computer program for control system design is going to be the coding dealing with input, output, and program sequence control.

Many computer programs in classical and a few in modern control theory now exist (Ref 1,2,3,4, and 5), but most can be quite cumbersome for the user. These programs are written in terms of computational and computer resource efficiency, often at the expense of user efficiency. User efficiency is defined as the capability of a user to get exactly what he needs from a program execution (no more and no less) with the minimum amount of work and expertise in the program operation.

There is no unique solution to a given control problem, and thus an iterative procedure with one or more design parameters may be required. This can best be achieved in an interactive environment where the user can make selective changes in the design parameters based upon intermediate outputs. The problem then is to develop an interactive, user oriented, program for the I-O (input-output) and program sequence control functions which can be put around any set of control system design theory algorithms.

The Problem

Professor Brian Porter of the University of Salford, England, has been developing a control system design theory based on discrete state variable feedback to achieve a time optimal response at any given sampling rate. The algorithms for this design process have been furnished by Professor Porter and the Air Force Flight Dynamics Lab in the computer program FORTRAC. This batch mode program which was written for the University of Salford computer and adapted to run on the Wright-Patterson AFB computer system along with the SCOPE operating system can be cumbersome for the user.

The problem focused on in this investigation is to develop the interactive I-O and program sequence control software for the design algorithms of FORTRAC which will give a user oriented design package that will operate efficiently on the CDC-6600/Cyber-74 computer system at Wright-Patterson AFB, Ohio. The capability of this package, INTERAC, is then demonstrated by synthesizing control laws for a fly-by-wire system for the longitudinal axis of the C-141 and F-4E aircraft.

Scope and Objectives

The software was designed to meet the following characteristics: (1) Input timing and format will be under user control. (2) Program termination will not be caused by any format errors in an input sequence. In case of such an error, the user will be notified and be asked to reenter that item. (3) Output content and format will be user defined. (4) Only user desired elements of the program need be executed and in user defined order. (5) Program directions will be printed for the beginning user or they can be suppressed for the experienced user.

(6) Additional algorithms can be added to the package with a minimum of additional programming. (7) The I-O subroutines can be used directly in any program requiring a user oriented, interactive environment. (8) The package will operate within the 60,000 word field length available to the interactive users on the CDC computers at Wright-Patterson AFB.

The control laws for the C-141 and F-4E fly-by-wire systems are obtained for the longitudinal axis and at only one flight condition.

Development

Chapter 2 gives an overview of the FORTRAC package with specific emphasis on the algorithms incorporated into INTERAC. The specialized INTERAC software is discussed in chapter 3. The mathematical models for the C-141 and the F-4E are developed in Chapter 4. The chapter concludes with a discussion of the control laws obtained for the two aircraft. Chapter 5 presents the conclusions and recommendations for inproving the software package and for continued control system development for the C-141 and F-4E.

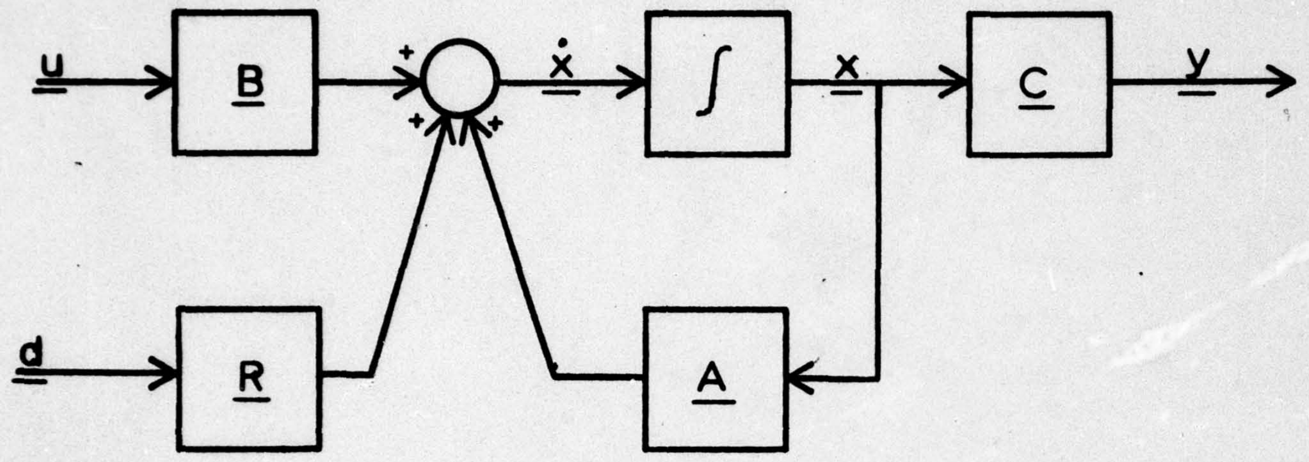
The actual FORTRAN Extended source code for the specialized INTERAC software is given in Appendix A, and a user's guide for the INTERAC package is contained in Appendix B. Appendix C is a programmer's guide for using the specialized INTERAC subroutines in other applications. Card sequence for the batch mode program, FORTRAC, is shown in Appendix D.

II. FORTRAC Computer Package

FORTRAC, a software package for the design of multivariable digital control systems, (Ref 6) was developed by Professor Brian Porter and furnished under contract to the Air Force Flight Dynamics Lab. This package is a collection of 42 subroutines tied together by a master program written for the particular design problem attempted. The master program furnished to the Flight Dynamics Lab will design a discrete control law for a sixth order or smaller system, design an observer, and run a time simulation of the system for the resulting controller. The theory is good for any order system, but a larger dimensioned master program would be required for higher order systems.

Table I shows the subroutines from FORTRAC grouped into four general functional categories. Most of the subroutines listed under control law synthesis and utility are included in INTERAC, and their specific function will be discussed in this chapter. A description of the other subroutines can be obtained from Reference 6.

FORTRAC is designed to take the continuous time description of a linear system (Fig 1), and to synthesize a control law for any of the following classes of problems: (1) discrete-time time-optimal regulator, (2) discrete-time time-optimal disturbance rejector, and (3) discrete-time time-optimal tracker.



$\underline{x}$ is an N dimensional state vector
$\underline{u}$ is an M dimensional control vector
$\underline{d}$ is an NDD dimensional disturbance vector
$\underline{y}$ is an NYY dimensional output vector
$\underline{A}$ is an N by N state matrix
$\underline{B}$ is an N by M control matrix
$\underline{C}$ is an NYY by N output matrix
$\underline{R}$ is an N by NDD disturbance matrix

Figure 1. Continuous Time System

Table I
FORTRAC Subroutines

Control Law Synthesis	Observer Design	Simulation	Utility
RPDATA	TRANFAC	DISCLS	ZEROELE
EIGSTR	FULLOB1	*EIGNVAL	DF01CKF
EIGVAV	REDOB1	DISSIMU	MATRE10
SAMPLE	REDOB2	GENCOM	MATPR10
AUGMAT		GENDIST	SETQUP
TRANFAB		*STSTATE	UREFORM
*CONGAIN		STIMEST	*ACHOROW
		SAMSIMU	SUB1000
		*KUTAMER	SUB2000
		DIF	SUB3000
		*STCONT	*APLFORM
		STIMECO	*BRNFORM
		PRIRES	*STQDOWN
		SAMRES	CINDS
			CALCBAR
			CONLAW

* These names had to be shortened from the original names in Reference 6 in adapting the package to operate on the CDC computer system at Wright-Patterson AFB.

The type of problem to be solved and the dimensions of the input system are handled by RPDATA. The I-O of the matrix data are handled by MATRE10 and MATPR10 respectively. These functions will be handled in INTERAC by the specialized INTERAC software.

Sampled Data Transformation

The first step in any class of problem is to transform the continuous system of Figure 1, described by Eq (1), to a sampled-data system description, Eq (2).

$$\begin{aligned} \dot{\underline{x}}(t) &= \underline{A}\,\underline{x}(t) + \underline{B}\,\underline{u}(t) + \underline{R}\,\underline{d}(t) \\ \underline{y}(t) &= \underline{C}\,\underline{x}(t) \end{aligned} \qquad (1)$$

$$\underline{x}(k+1) = \underline{F}(T)\underline{x}(k) + \underline{G}(T)\underline{u}(k) + \underline{DR}(T)\underline{d}(k)$$
$$\underline{y}(k) = \underline{C}\underline{x}(k) \tag{2}$$

All discrete functions of k are implied to be functions of kT. Two major subroutines are used in this operation. The first is EIGSTR, which also calls EIGVAV. The eigenvalues and eigenvectors of the continuous system are determined with the aid of matrix manipulation routines from a common user mathematics library. The description of the matrix manipulation routines required for FORTRAC and INTERAC are given later in this chapter.

The second major subroutine used is SAMPLE, which generates the discrete matrices $\underline{F}(T)$, $\underline{G}(T)$, and $\underline{DR}(T)$ from $\underline{A}$, $\underline{B}$, and $\underline{R}$. The transformation can be accomplished using Eqs (3),(4), and (5).

$$\underline{F}(T) = e^{\underline{A}T} \tag{3}$$

$$\underline{G}(T) = \int_0^T e^{\underline{A}t}\,\underline{B}\,dt \tag{4}$$

$$\underline{DR}(T) = \int_0^T e^{\underline{A}t}\,\underline{R}\,dt \tag{5}$$

In general, an exact solution to the above equations cannot be easily determined. To facilitate the solution the diagonal matrix $\underline{\Lambda}$ which satisfies Eq (6) is produced,

$$\underline{A} = \underline{U}\,\underline{\Lambda}\,\underline{U}^{-1} \tag{6}$$

where $\underline{U}$ is the matrix of eigenvectors obtained from EIGSTR. At this point the subroutine SAMPLE is limited to systems which have non-repeated eigenvalues. Multiple eigenvalues would produce off diagonal terms in $\underline{\Lambda}$, thus this simplification could not be used. The equations for the discrete transformation reduce to (Ref 6:27)

$$\underline{F}(T) = \underline{U}\, e^{\Delta T}\, \underline{U}^{-1} \tag{7}$$

$$\underline{G}(T) = \underline{U}\left[\int_0^T e^{\Delta t}\, dt\right]\underline{U}^{-1}\underline{B} \tag{8}$$

$$\underline{DR}(T) = \underline{U}\left[\int_0^T e^{\Delta t}\, dt\right]\underline{U}^{-1}\underline{R} \tag{9}$$

The computation of $e^{\Delta T}$ becomes a set of scalar problems $e^{\lambda_i T}$, where the λ_i's are the eigenvalues of $\underline{A}$ which appear on the diagonal of $\underline{\Lambda}$. The constant matrices $\underline{U}$, $\underline{U}^{-1}$, $\underline{B}$, and $\underline{R}$ are taken outside the the integrals, and the integrals become sets of scalar integrations. The $\underline{C}$ matrix remains the same in the transformation from continuous to sampled-data form.

Augmentation

In the disturbance rejection and tracking problems, the next step is to augment the system with discrete-time integrators shown in Eq (10).

$$\underline{z}(k+1) = \underline{z}(k) + T\left[\underline{e}'(k)\right] \tag{10}$$

where the error, $\underline{e}'(k)$, for the disturbance rejector is $\underline{y}(k)$ and for the tracker is $\underline{y}(k)-\underline{v}(k)$. The rationale and specific requirements for the use of integral feedback is given in Reference 7 (papers 7 and 9). The command input matrix, $\underline{DE} = \underline{IT}$, is established for the input command, v(k), in the augmented system equation. The augmented discrete system is then represented by Eq (11).

$$\underline{x}'(k+1) = \underline{AF}\,\underline{x}'(k) + \underline{AG}\,\underline{u}(k) + \underline{ADR}\,\underline{d}(k) + \underline{ADE}\,\underline{v}(k) \qquad (11)$$

where

$$\underline{AF} = \begin{bmatrix} \underline{F} & \underline{\emptyset} \\ \underline{CT} & \underline{I} \end{bmatrix} \qquad \underline{AG} = \begin{bmatrix} \underline{G} \\ \underline{\emptyset} \end{bmatrix}$$

$$\underline{ADR} = \begin{bmatrix} \underline{DR} \\ \underline{\emptyset} \end{bmatrix} \qquad \underline{ADE} = \begin{bmatrix} \underline{\emptyset} \\ -\underline{DE} \end{bmatrix}$$

$$\underline{x}'(k) = \begin{bmatrix} \underline{x}(k) \\ \underline{z}(k) \end{bmatrix} \qquad (12)$$

Control Law Synthesis

The control law to be synthesized is given by Eq (13).

$$\underline{u}(k) = \underline{K}\,\underline{x}'(k) \qquad (13)$$

The first step uses subroutine TRANFAB to transform the matrices

AF and AG into Brunovsky controllable canonical form. This is done using the first method of Aplevich (Ref 8:124-126). The utility subroutines SETQUP through STQDOWN (Table I) are used to achieve the transformation and to generate the transformation matrix $\underline{T}^{-1}$ to satisfy Eq 14.

$$\underline{x}''(k) = \underline{T}^{-1}\underline{x}'(k) \tag{14}$$

The control law synthesized to meet Eq (13) is as follows.

$$\underline{u}(k) = \underline{L}^{-1}[\underline{H} + \underline{E}]\,\underline{T}^{-1}\underline{x}'(k) \tag{15}$$

The $\underline{H}$ and $\underline{L}$ matrices are formed from the Brunovsky controllable canonical form of AF and AG respectively (Ref 6:44-46). The $\underline{E}$ matrix is used to specify the desired closed-loop eigenvalues. For the time-optimal control law the closed-loop eigenvalues are placed at the origin in the z-plane and the $\underline{E}$ matrix is the zero matrix. Refer to Reference 6, page 45, for the algorithm used for calculating $\underline{E}$ for non-zero eigenvalues.

Matrix Manipulation Routines

Table II shows the matrix manipulation routines called in the original FORTRAC package and the corresponding routines used in the version adapted to run on the CDC computers at Wright-Patterson AFB. The eigenstructure analysis routines were taken from the EISPACK Library (Ref 9) available on the CDC computers. The other routines were written by the author and the source code is shown in Appendix A.

Table II
Matrix Manipulation Routines

Routines Called In The Original FORTRAC Package		Routines Used In The Wright-Patterson Adaptation
F01CAF		PRESET
F01CBF		IDENT
F01CCF		COPYAB
F01CDF		ADDMAT
F01CEF		SUBMAT
F01CJF		TRANPOS
F04AEF		INVERT
F01ATF	Eigenstructure Routines	BALANC
F01AKF		ELMHES
F02AGF		ELTRAN
F02APF		HQR
		HQR2
		BALBAK

Appendix C gives the syntax required to use the author's matrix manipulation routines and gives a brief description of the algorithms used.

The above routines along with DISCLS are included in the INTERAC package. The subroutine DISCLS calculates the closed-loop equations which can then be used in a discrete simulation.

III. Specialized INTERAC Software

Before adapting FORTRAC to an interactive program, several interactive programs (Ref 1,2,3,4, and 5) available on the Wright-Patterson AFB computer were examined to see how efficient they were for the user. The first item of note was that each program had different input format, output content and format, and control structure. This greatly increases the workload on the user who has to use several programs in a design effort. With the exception of OPTCON, there are no provisions for correcting an erroneous data point entry without reentering the entire data set. All programs are terminated by the SCOPE operating system for a simple input error by the user. All programs allowed the user to select the options he wished to execute, but there are no provisions to stop between options and allow the user to decide if the sequence should be continued. In general each program has some good and many bad points in regard to user efficiency.

Three methods of developing FORTRAC into an efficient, user oriented, interactive, package were considered. The first was to redefine the operating system for interactive computer use. Although this would provide the greatest improvement for all classes of interactive programs, the time and resources required would be beyond the capability of this investigation. The second method would be to utilize a higher order computer language other than FORTRAN. Of the languages available on the Wright-Patterson CDC-6600/Cyber-74 computer system, FORTRAN Extended seemed to be the best suited for the job. This left direct FORTRAN programming, possibly sacrificing a little computer resource efficiency in order to achieve the desired user efficiency.

The INTERAC software described in this chapter was developed to meet the specifications given in Chapter 1 using the maximum capabilities of the FORTRAN Extended language (Ref 10) and the SCOPE operating system (Ref 11) available on the Wright-Patterson AFB computers. The three areas in a typical program operation cycle addressed are: (1) input, (2) output, and (3) program sequence control.

Input includes both numeric data and alphanumeric commands. The two subroutines developed for this are READNUM (read numeric) and READCOM (read command). The data output is controlled in content by an entry in a variable output table and in format by the subroutines PRINTR and LISTER. The program sequence control is based on entries in a valid command table and is achieved in this program with the subroutine COMND.

Input Subroutines

The subroutine READNUM was first conceptualized due to the deficiencies in the SCOPE operating system on the Wright-Patterson AFB computer. When a format error is made in a numeric input operation, an error message is printed and the program is terminated. This can be quite frustrating to a user who is just entering the last number of a 10 by 10 matrix. The READNUM subroutine reads all inputs as alphanumeric characters and checks for format errors. If any errors are detected the user is asked to reenter those items. The error checking is based on a legal entry table to be discussed later.

Table III
READNUM, Legal Entry Table

◆A	◆B	◆C	LEGAL ENTRY TABLE	◆A	◆B	◆C	LEGAL ENTRY TABLE
◆◆	◆◆	01	66666666666666666666				
:	00	02	11111111111153555555	5	40	34	11111111111113111112
A	01	03	11111111111131555553	6	41	35	11111111111113111112
B	02	04	11111111111151555555	7	42	36	11111111111113111112
C	03	05	11111111111131555553	8	43	37	11111111111113111112
D	04	06	11111111111151555555	9	44	38	11111111111113111112
E	05	07	11111111111151555225	+	45	39	11111111111113551552
F	06	08	11111111111151555555	−	46	40	11111111111113551552
G	07	09	11111111111151555555	◆	47	41	11111111111113555552
H	10	10	11111111111151555555	/	50	42	11111111111142455554
I	11	11	11111111111151555555	(	51	43	11111111111153555555
J	12	12	11111111111151555555	)	52	44	11111111111153555555
K	13	13	11111111111151555555	$	53	45	11111111111135777773
L	14	14	11111111111131555553	=	54	46	11111111111153555555
M	15	15	11111111111151555555		55	47	11111111111112244441
N	16	16	11111111111151555555	,	56	48	11111111111112344441
O	17	17	11111111111151555555	.	57	49	11111111111113555532
P	20	18	11111111111151555555	#	60	50	11111111111153555555
Q	21	19	11111111111151555555	[	61	51	11111111111153555555
R	22	20	11111111111151555553	]	62	52	11111111111153555555
S	23	21	11111111111151555555	%	63	53	11111111111153555555
T	24	22	11111111111151555555	"	64	54	11111111111153555555
U	25	23	11111111111151555555	_	65	55	11111111111153555555
V	26	24	11111111111151555555	!	66	56	11111111111153555555
W	27	25	11111111111151555555	&	67	57	11111111111153555555
X	30	26	11111111111151555555	'	70	58	11111111111153555555
Y	31	27	11111111111151555555	?	71	59	11111111111153555555
Z	32	28	11111111111131555553	<	72	60	11111111111153555555
0	33	29	11111111111113111112	>	73	61	11111111111153555555
1	34	30	11111111111113111112	@	74	62	11111111111153555555
2	35	31	11111111111113111112	\	75	63	11111111111153555555
3	36	32	11111111111113111112	^	76	64	11111111111153555555
4	37	33	11111111111113111112	;	77	65	11111111111153555555

◆A -- CHARACTER
◆B -- OCTAL VALUE OF CHARACTER REPRESENTATION
◆C -- DECIMAL INDEX FOR TABLE
◆◆ -- THIS ENTRY IN THE TABLE PROVIDES FOR AN END OF LINE INDICATOR.

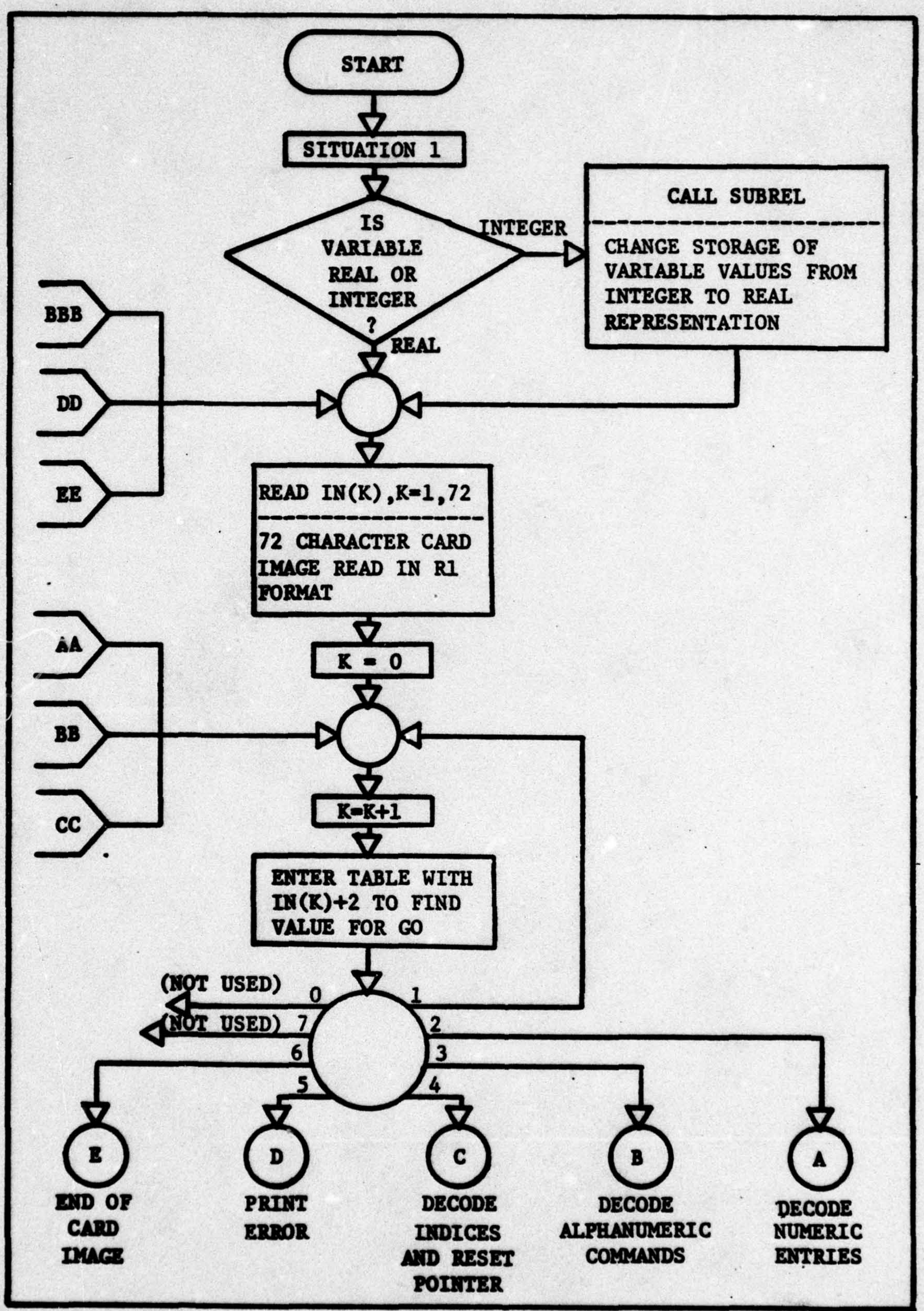

START
SITUATION 1
IS VARIABLE REAL OR INTEGER ?
INTEGER
REAL
CALL SUBREL
CHANGE STORAGE OF VARIABLE VALUES FROM INTEGER TO REAL REPRESENTATION
BBB
DD
EE
READ IN(K),K=1,72
72 CHARACTER CARD IMAGE READ IN R1 FORMAT
K = 0
AA
BB
CC
K=K+1
ENTER TABLE WITH IN(K)+2 TO FIND VALUE FOR GO
(NOT USED) 0
1
(NOT USED) 7
2
6
3
5
4
E
END OF CARD IMAGE
D
PRINT ERROR
C
DECODE INDICES AND RESET POINTER
B
DECODE ALPHANUMERIC COMMANDS
A
DECODE NUMERIC ENTRIES

Figure 2. READNUM Flow Chart, Part I

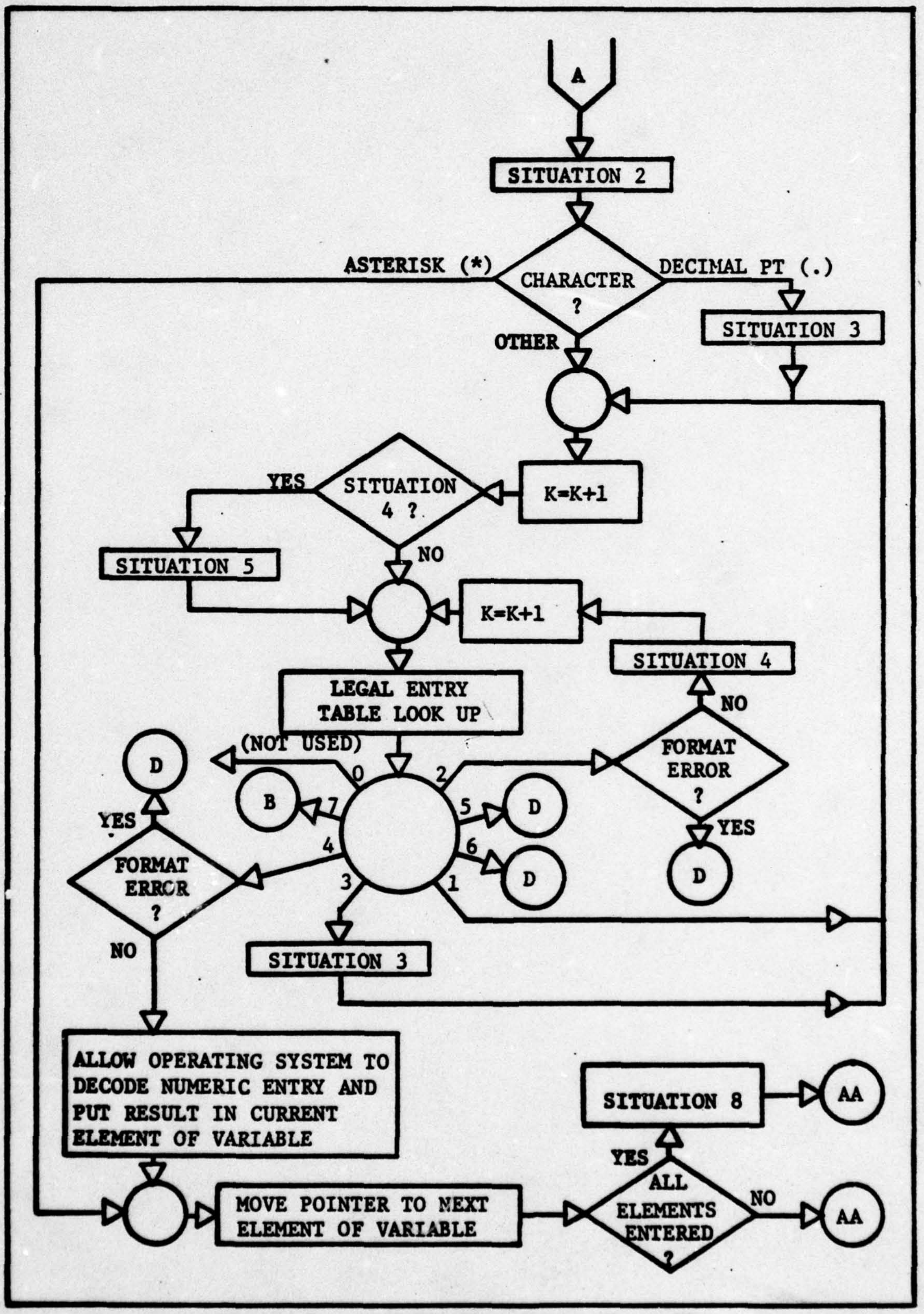


Figure 3. READNUM Flow Chart, Part II

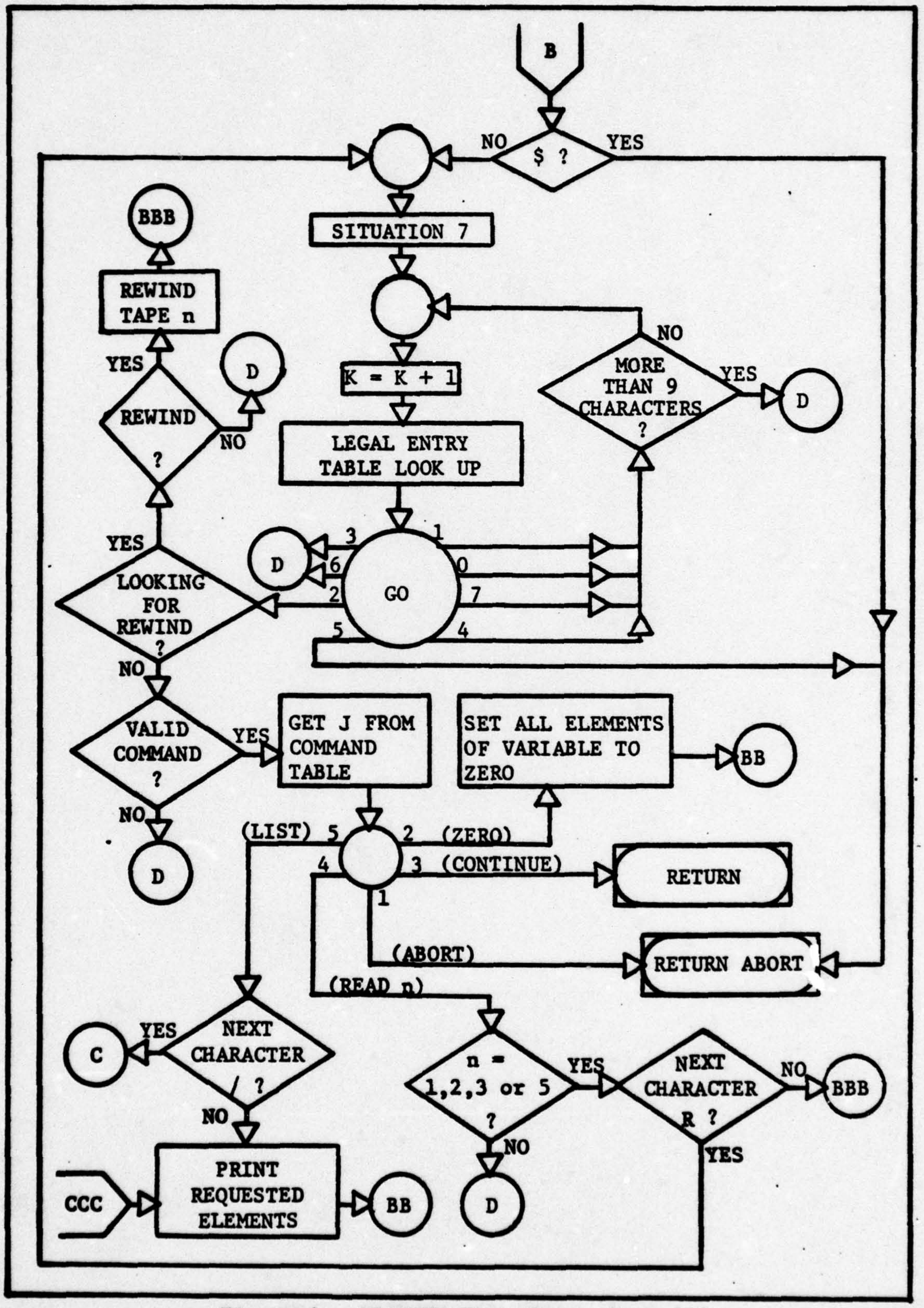


Figure 4. READNUM Flow Chart, Part III

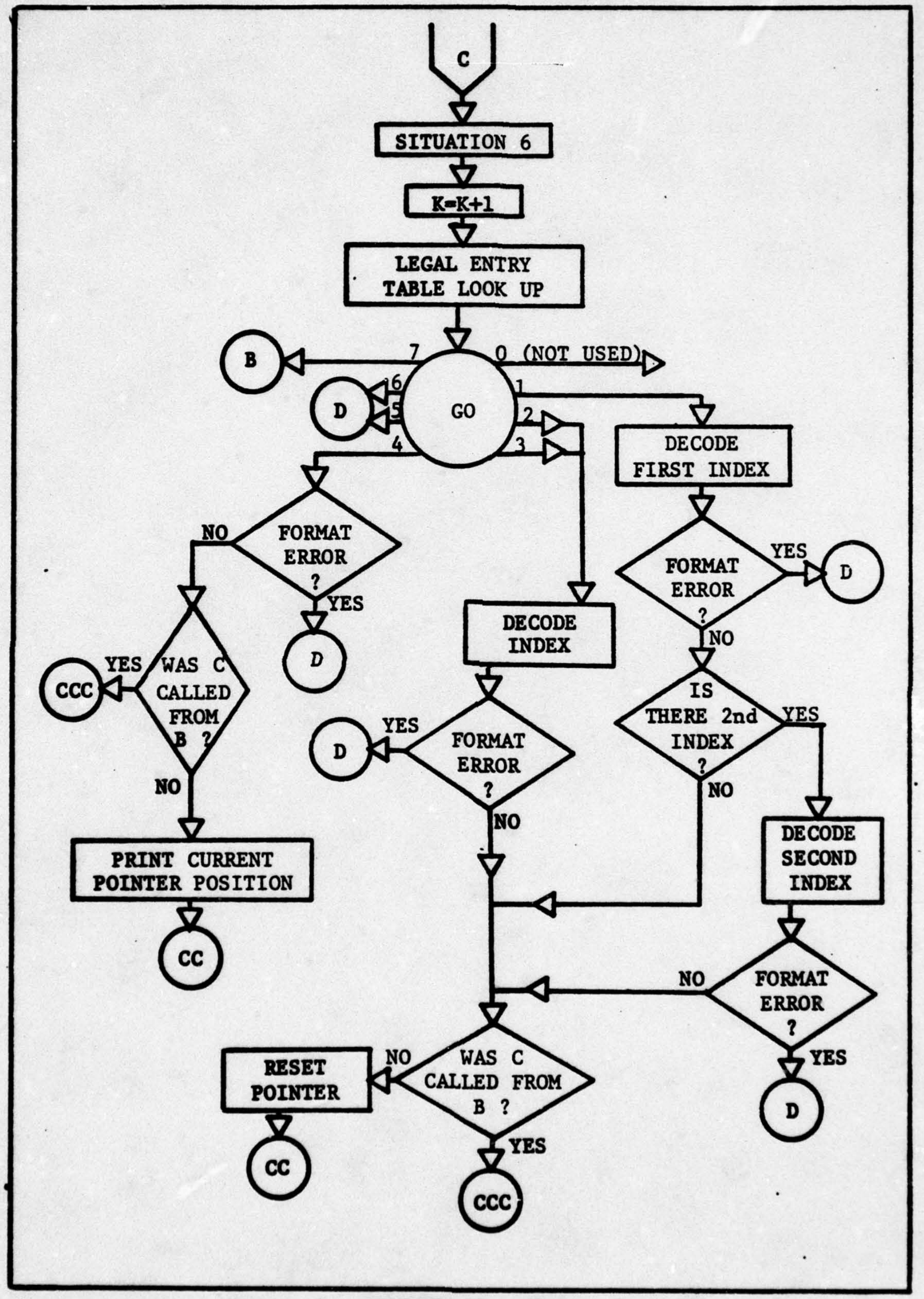


Figure 5. READNUM Flow Chart, Part IV

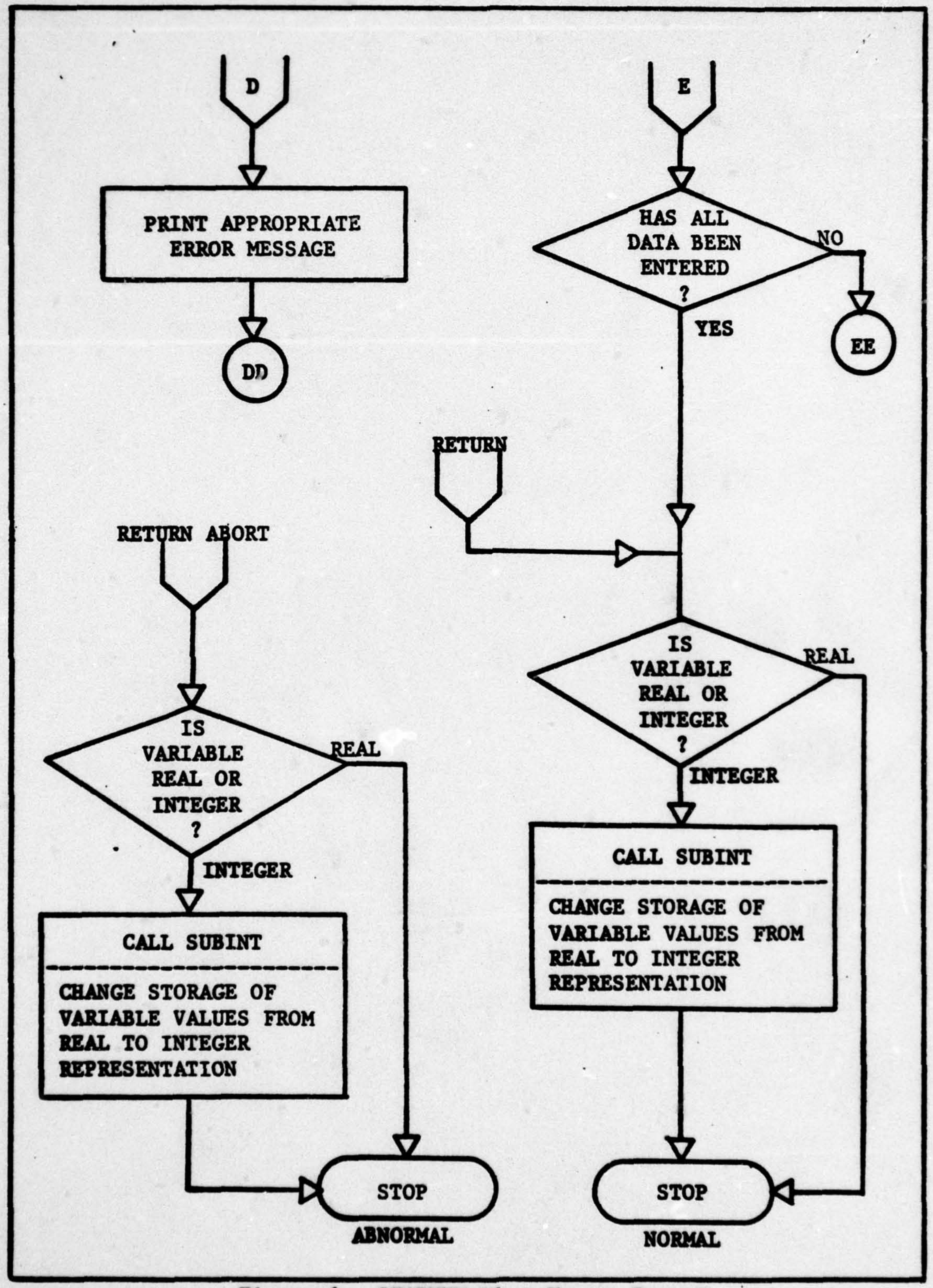

D
PRINT APPROPRIATE ERROR MESSAGE
DD
E
HAS ALL DATA BEEN ENTERED ?
NO
EE
YES
RETURN
RETURN ABORT
IS VARIABLE REAL OR INTEGER ?
REAL
INTEGER
CALL SUBINT
CHANGE STORAGE OF VARIABLE VALUES FROM REAL TO INTEGER REPRESENTATION
STOP
ABNORMAL
IS VARIABLE REAL OR INTEGER ?
REAL
INTEGER
CALL SUBINT
CHANGE STORAGE OF VARIABLE VALUES FROM REAL TO INTEGER REPRESENTATION
STOP
NORMAL

Figure 6. READNUM Flow Chart, Part V

The power in READNUM is particularly evident in matrix data entry. Each element of the matrix is constantly available for data entry or change. A data entry error noted after a carriage return can still be corrected on the next line. Six special commands give the user complete control over the data input process.

The basic flow chart for READNUM is shown in Figure 2. As can be seen, this subroutine can handle variables in real or integer representation. All data within the subroutine are represented as real numbers, and just prior to termination the subroutine converts them back to integer form if necessary.

All input characters are checked against the legal entry table, Table III. This table is stored as an array of 65 computer words. Each octal bit of the word is used in a different situation. Thus each character can cause one of 8 actions in up to 20 different situations. Situation 1 (the table is read from right to left) is used at the first decision point shown in Figure 2. The remainder of the flow chart for the READNUM subroutine is shown in Figures 3 thru 6.

All matrix elements are entered by rows in free format until all elements have been filled, unless the sequence is altered with one of the special commands. The first special character of interest is the asterisk (*), shown in Figure 3. The entry of the * causes the data entry pointer to move to the next element of the matrix with no change to the value of the current element. Figure 4 shows the section for decoding the six special alphanumeric commands and the special character dollar sign ($). Any time the $ is encountered a return abort is executed from the subroutine.

The six alphanumeric commands are "abort","zero","continue", "read","list", and "/n,m/". The "abort" command has the same result as the $. "Zero" causes all elements of the variable to be set to zero and is useful for entering sparse matrices. "Continue" results in a normal termination of the subroutine regardless of the data entry pointer position. "Read" allows the input data to be read from the file TAPE1, TAPE2, or TAPE3. An optional parameter following the command "read n" is "rewind", which causes TAPEn to be rewound prior to reading data from it. The command "list" causes the current contents of the variable elements to be printed at the terminal. The optional parameter "/n,m/" following "list" is used to print only one element, one row, or one column of the variable. All of the above commands may be abbreviated to as little as the first letter. The command "/n,m/" is used to reset the data entry pointer to element (n,m). This allows previous errors to be corrected or several elements that need not be entered to be skipped.

The technical aspects for using this and other INTERAC subroutines for different programs can be found in the Programmer's Guide, Appendix C.

The second input subroutine developed is READCOM. This subroutine reads alphanumeric commands of 9 characters or less and checks them against a list of valid commands. Commands may have any characters, including single embedded blanks, except the slash (/), comma (,), and dollar sign ($). Any abbreviation that is unique in the command list may be used. The flow chart for READCOM is shown in Figure 7. Multiple commands may be entered if they are separated by "/" or at least two blanks. A command string is also recognized and identified. A command string is a command followed by one or more other commands

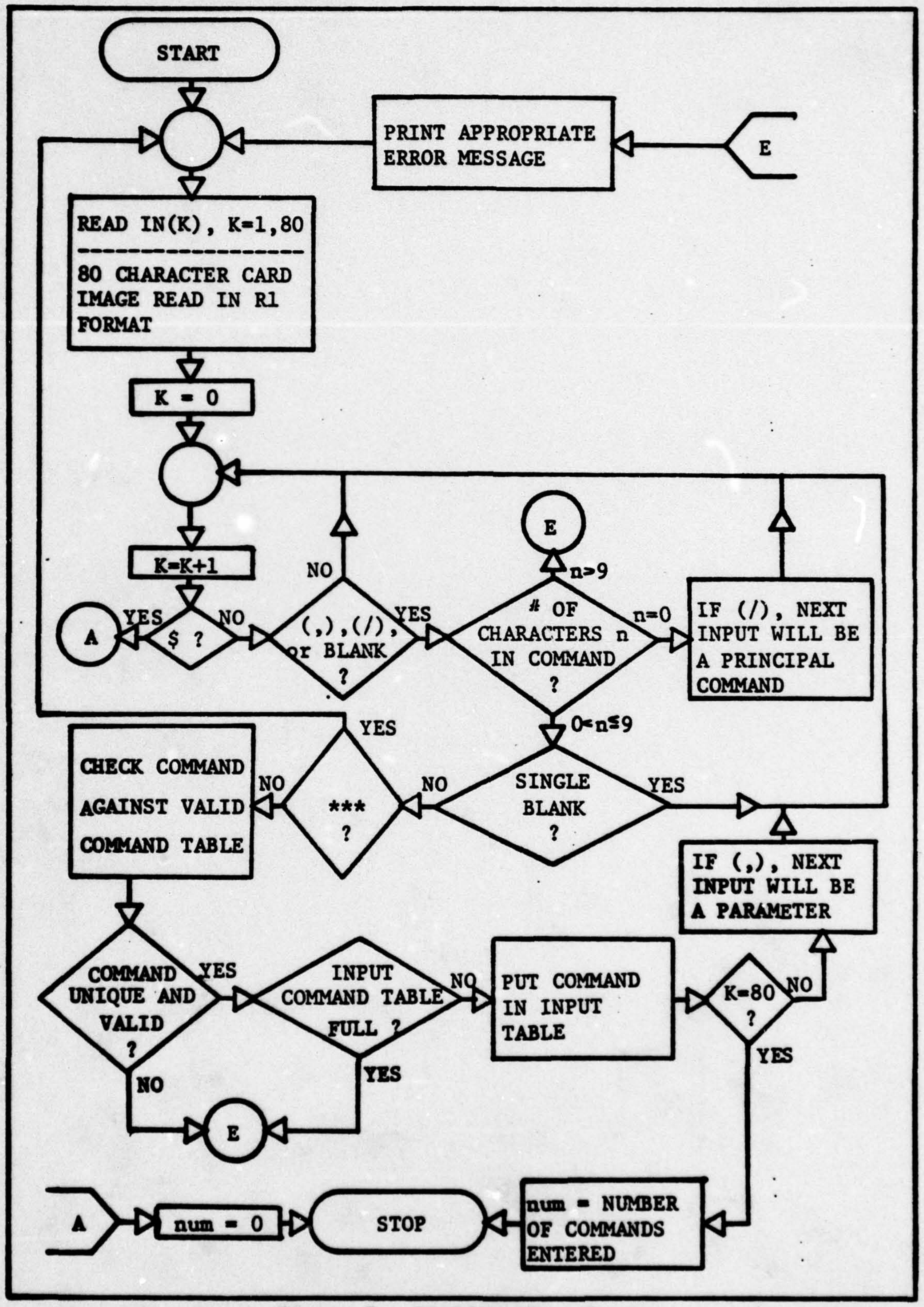

START
PRINT APPROPRIATE ERROR MESSAGE
E
READ IN(K), K=1,80
80 CHARACTER CARD IMAGE READ IN R1 FORMAT
K = 0
K=K+1
$?
YES
NO
A
(,),(/), or BLANK ?
NO
YES
E
n>9
# OF CHARACTERS n IN COMMAND ?
n=0
IF (/), NEXT INPUT WILL BE A PRINCIPAL COMMAND
0<n≤9
SINGLE BLANK ?
YES
NO
*** ?
YES
NO
CHECK COMMAND AGAINST VALID COMMAND TABLE
IF (,), NEXT INPUT WILL BE A PARAMETER
COMMAND UNIQUE AND VALID ?
YES
NO
INPUT COMMAND TABLE FULL ?
NO
YES
PUT COMMAND IN INPUT TABLE
K=80 ?
NO
YES
E
A
num = 0
STOP
num = NUMBER OF COMMANDS ENTERED

Figure 7. READCOM Flow Chart

acting as modifing parameters. The parameters are separated from the command and from each other by commas.

There are two special commands that are used internally in the subroutine. The $ causes a return abort. Anytime the $ is encountered the subroutine returns with an indication that no commands have been entered. The second special command is "***". READCOM normally reads only one 80 character input line. But when "***" is encountered as a command or parameter, another line is read. There are four error conditions recognized by READCOM. After the error message is printed the user is asked to reenter and is allowed another 80 character line.

Output Subroutines

Five of the legal commands interpreted by READCOM are used to change parameters in the output subroutines. Three of these commands are concerned with changing entries in the variable output table. Data output is sent to two possible locations as prescribed by the user with these commands. Data can be displayed at the terminal or can be written on a local file (to be disposed to a line printer after the program is terminated), or both.

The subroutine PRINTR is called by the computational subroutines whenever the value of a variable is changed. In this subroutine the variable output table is checked to see where the variable values should be printed. If display at the terminal or write to file are indicated the subroutine LISTER is called. Within LISTER the data is formated to appear on a 72 character line for the terminal or a 132 character line for the line printer output as appropriate. The number of significant digits used in each case can be specified by the user.

Program Sequence Control

The output format and content as well as all program operations are controlled by the user's alphanumeric inputs. Table IV lists all legal commands available to the user.

Table IV

Legal Commands For The INTERAC Package

Utility Options	I-O Options	Design Options	Variables			
OPTIONS	ENTER	RUN	AMATRIX	AFMATRIX	CEIGEN	INTEGRATE
VARIABLES	CHANGE	FORTRAC	BMATRIX	AGMATRIX	DEIGEN	MODAL
SAVE	DFORMAT	SAMPLE	RMATRIX	AEMATRIX	DESIGEN	INVMODAL
STOP	OFORMAT	AUGMAT	CMATRIX	ARMATRIX	CLEIGEN	UREFORM
END	DISPLAY	TRANSFORM	FMATRIX	ACMATRIX	TSAMPLE	APLFORM
RESTART	OUTPUT	CONLAW	GMATRIX	BFMATRIX	DIMENSION	BRNFORM
REWIND	DELETE	CLOSELOOP	DEMATRIX	BGMATRIX	STATES	CINDICES
	PRINT	SETDIMEN	DRMATRIX	TINVERSE	CONTROLS	COFMATRIX
				KMATRIX	COMMANDS	
				CLMATRIX	DISTURBS	

Refer to Appendix B for the description and use of each command. The table of legal commands is stored in the array LIST, dimensioned n by 7. Column 1 of LIST contains the Hollerith representation of each option or variable name.

The other columns of LIST contain all the information needed to decide what action to take when the command is input by the user.

Column 2 divides the commands into four types: (1) option names, (2) real variable names, (3) integer variable names, and (4) output variable names. The output variables are intermediate results in a computational algorithm. These results are stored in temporary scratch locations in memory, and the user may not enter or change data within them. The user may indicate if and where the variable results should be printed when they are calculated. All real and integer variables are stored in contiguous locations in a common block. The other five columns of LIST indicate the starting location for the variable within the common block, the maximum row and column dimensions allowed, and the currently used row and column dimensions. Option names are divided into five catagories by an entry in the last column of LIST. These catagories are shown in Table V.

Table V
Option Catagories

Entry In Column 7 Of LIST	Description
+n	Principal command only, with n required parameters
0	Principal command only, parameters optional
-1	Principal command only, no parameters allowed
-2	Principal command or parameter, no parameters allowed
-3	Parameter only

Subroutine COMND uses the entries in the array LIST to initiate the action requested by the user's command input. Refer to Appendix A for the listing of this subroutine.

IV. C-141 and F-4E Mathematical Modeling

An analog fly-by-wire control system for the C-141 has been designed and built by Honeywell, Inc. under contract to the Air Force Flight Dynamics Lab (Ref 12). The results of this project could provide data against which to check a digital control system designed using INTERAC. The second plant chosen to test the INTERAC package is an F-4E (Ref 13). This chapter describes the mathematical modeling process for the two plants. After several computer runs, the C-141 was dropped from consideration due to modeling problems to be discussed later. The chapter ends with a discussion of the results obtained for the F-4E.

C-141 Model

The model for the C-141 was derived using data from Reference 12. The longitudinal equations of motion are given as

$$\ddot{\theta} = M_{\dot{w}}\dot{w} + M_w w + M_q\dot{\theta} + M_{\delta e}\delta e + M_{\delta sp}\delta sp \tag{16}$$

$$\dot{w} = Z_u u + Z_w w + U_o\dot{\theta} + Z_{\delta e}\delta e + Z_{\delta sp}\delta sp \tag{17}$$

$$\dot{u} = X_u u + X_w w - g\theta + X_{\delta sp}\delta sp \tag{18}$$

$$\dot{h} = -w + U_o\theta \tag{19}$$

$$a_{z_{pilot}} = \dot{w} + U_o\dot{\theta} - l_x\ddot{\theta} \tag{20}$$

The coefficients for the heavyweight landing configuration are given in Table VI (Ref 12:28,31).

Table VI
C-141 Longitudinal Dimensional Derivatives

257,000 lb Weight, Gear Down, 119 KCAS at Sea Level

Parameter	Value	Parameter	Value	Parameter	Value
$M_{\dot{w}}$	-.00102	Z_u	-.321	X_u	-.044
M_w	-.0051	Z_w	-.558	X_w	.083
M_q	-.645	U_0	201	g	32.2
$M_{\delta e}$	-.743	$Z_{\delta e}$	-5.36	$X_{\delta sp}$	-4.25
$M_{\delta sp}$	.240	$Z_{\delta sp}$	33.0	l_x	47
				U_{co}	350

The transfer functions for the elevator and spoiler actuator-servo combinations were modeled as given in Eq 21 and 22 respectively.

$$\frac{\delta e}{e_i} = \frac{10}{s+10} \qquad (21)$$

$$\frac{\delta sp}{sp_i} = \frac{5}{s+5} \qquad (22)$$

These transfer functions and the above equations of motion were modeled as state equations in the form of Eq (1).

$$\begin{aligned} \underline{x}(t) &= \underline{A}\,\underline{x}(t) + \underline{B}\,\underline{u}(t) + \underline{R}\,\underline{d}(t) \\ \underline{y}(t) &= \underline{C}\,\underline{x}(t) \end{aligned} \qquad (1)$$

where

$$\underline{x}(t) = \begin{bmatrix} \dot{\theta}(t) \\ \theta(t) \\ w(t) \\ u(t) \\ h(t) \\ \delta e(t) \\ \delta sp(t) \end{bmatrix} \qquad \underline{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 10 & 0 \\ 0 & 5 \end{bmatrix} \qquad \underline{u}(t) = \begin{bmatrix} e_i \\ sp_i \end{bmatrix} \qquad \underline{R} = \emptyset$$

$$A = \begin{bmatrix} -.85 & 0 & -.00453 & .00033 & 0 & -.736 & .206 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 201 & 0 & -.558 & -.321 & 0 & -5.36 & 33 \\ 0 & -32.2 & .083 & -.044 & 0 & 0 & -4.25 \\ 0 & 201 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -5 \end{bmatrix} \qquad (23)$$

The C matrix was then chosen to give a C^* output. C^* is described in Ref 14 and the equation used is (Ref 12:55)

$$C^* = a_{z_{pilot}} + K_q \dot{\theta} \qquad (24)$$

$$K_q = .217 \quad \text{for} \quad U_{co} = 350$$

This was the first hint that the data provided by Ref 12 may not be valid. Using the definition of C^* from Ref 14, the value of K_q should have been 10.8. The C matrix then became.

$$C = [\,1 \;\; 0 \;\; -.345 \;\; -.336 \;\; 0 \;\; 29.3 \;\; 23.3\,] \qquad (25)$$

where β is 442.2 or 452.8. The control law for a discrete-time time-optimal tracker was synthesized for both $\underline{C}$ matrices with similar results. The gains were excessive (greater than 10^4) and a time simulation showed huge overshoots and impossible control requirements (elevator deflections of 400°). A modeling problem was suspected when it was noted that the calculated Brunovsky transformation matrix $\underline{T}^{-1}$ was incorrect. This was due to computer numerical problems caused by the fact that the plant as modeled was almost uncontrollable. In an attempt to verify the modeling, the coefficients taken from Page 28 and 31 of Ref 12 were checked against the analog flow chart on Page 33 of Ref 12. There were numerous numerical differences and a sign difference in the δ_{sp} term in the $\dot{u}$ equation (Eq 18). Thus the validity of any comparison of a discrete fly-by-wire system developed using INTERAC/FORTRAC with the analog system described in Ref 12 was in serious doubt. At this time it was decided to attempt a control system design for another aircraft.

F-4E Model

The F-4E was modeled using data from a lab project from the EE 6.41 course at AFIT (Ref 13). The coefficients of Table VII were used in the following equations of motion:

$$\ddot{\theta} = \frac{1}{\left(\frac{I_y}{Sqc}\right)} \left[\frac{c}{2U_0} C_{m_{\dot{\alpha}}} \dot{\alpha}' + C_{m_\alpha} \alpha' + \frac{c}{2U_0} C_{m_q} \dot{\theta}' + C_{m_{\delta e}} \delta e \right] \tag{26}$$

$$\dot{\alpha}' = \frac{1}{\left(\frac{mU_o}{Sq} - \frac{c}{2U_o} C_{Z_{\dot{\alpha}}}\right)} \left[C_{Z_u} u' + C_{Z_\alpha} \alpha' + \left(\frac{mU_o}{Sq} + \frac{c}{2U_o} C_{Z_q}\right) \dot{\theta}' + C_{Z_{\delta_e}} \delta e \right] \qquad (27)$$

$$\dot{u}' = \frac{1}{\left(\frac{mU_o}{Sq}\right)} \left[C_{X_u} u' + C_{X_\alpha} \alpha' + C_W \theta' + C_{X_{\delta_T}} \delta_T \right] \qquad (28)$$

Table VII
F-4E Longitudinal Non-dimensional Derivatives

33,439 lb Weight, Gear Down, 138 KCAS at Sea Level

Parameter	Value	Parameter	Value	Parameter	Value
I_y	137,710	$C_{m_{\dot{\alpha}}}$	-.9801	C_{z_u}	-2.0
U_o	230.5	C_{m_α}	-.07634	C_{z_α}	-2.967
q	63.14	C_{m_q}	-2.066	C_{z_q}	-2.276
c	16.04	$C_{m_{\delta_e}}$	-.5776	$C_{z_{\delta_e}}$	-.3887
g	32.174	C_{x_u}	-.339	$C_{x_{\delta_T}}$	.2
S	530	C_{x_α}	.630		

Dimensionalizing and other relationships

$m = w/g$ $\quad C = -mg/Sq \quad u' = u/U_o \quad \alpha = 57.3\alpha' \quad \theta = 57.3\theta'$

The elevator actuator and servo were modeled as the first order lag

$$\frac{\delta e}{e_i} = \frac{20}{s + 20} \qquad (29)$$

and the engine and throttle actuator were modeled as

$$\frac{\delta_T}{T_i} = \frac{2}{s+2} \tag{30}$$

The statespace equation for the F-4E then became

$$\begin{aligned} \dot{\underline{x}}(t) &= \underline{A}\,\underline{x}(t) + \underline{B}\,\underline{u}(t) + \underline{R}\,\underline{d}(t) \\ \underline{y}(t) &= \underline{C}\,\underline{x}(t) \end{aligned} \tag{1}$$

where

$$\underline{x}(t) = \begin{bmatrix} \dot{\theta}(t) \\ \theta(t) \\ \alpha(t) \\ u(t) \\ e(t) \\ T(t) \end{bmatrix} \quad \underline{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 20 & 0 \\ 0 & 2 \end{bmatrix} \quad \underline{u}(t) = \begin{bmatrix} e(t) \\ T(t) \end{bmatrix} \quad \underline{R} = \underline{\emptyset}$$

$$\underline{A} = \begin{bmatrix} -.4115 & 0 & -.2424 & .009231 & -2.244 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ .989 & 0 & -.4146 & -.06946 & -.05431 & 0 \\ 0 & -.5717 & .3541 & -.04737 & 0 & 6.442 \\ 0 & 0 & 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 \end{bmatrix} \tag{31}$$

The $\underline{C}$ matrix was chosen to allow tracking of flight path angle, $\theta - \alpha$, as would be desirable in an automatic approach.

$$\underline{C} = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 & 0 \end{bmatrix} \tag{32}$$

A control law was synthesized to track flight path angle for various sampling times.

Results of F-4E Control Law Synthesis

At all sampling times a simulation showed the F-4E to be on track within 4 sample periods. This requires higher gains and higher control power as the sampling interval decreases. Table VIII shows some of the representative data obtained for sampling intervals of .05 to 5.0 seconds, with a unit step command of 1 degree.

Table VIII
Simulation Data for the F-4E with a Discrete-Time Flight Path Angle Tracking Control System

	Maximum				Final
Sampling Interval Seconds	Elevator Deflection	Change in Forward Velocity ft/sec	Negative Peak in Commanded Output	Magnitude of Feedback Gain Required	Change in Forward Velocity ft/sec
.05	-8010 °	-255	-15.6°	2.5×10^5	-255
.1	-1245 °	-154	- 4.9°	1.4×10^4	-154
.25	- 87.6°	- 62	- .7°	353	- 62
.5	- 12 °	- 29	- .1°	24	- 29
.6	- 7.3°	- 24	- .01°	15	- 24
.7	.4°	16	0.0	17.5	3.6
.8	.4°	14	0.0	12.5	3.2
.9	.3°	13	0.0	9.3	3.1
1.0	.3°	11	0.0	7.2	2.9
2.0	.2°	5	0.0	1.2	.08
5.0	.3°	13	0.0	.7	13.4

At sample times below .5 seconds, the elevator deflection required is beyond the capability of the aircraft. A sample time of 2 seconds requires the least control power and produces the smoothest response. But 8 seconds to track a 1 degree flight path change is excessive for aircraft control. Even the lowest practical sampling time of .5 seconds, may be too high for this fighter type aircraft.

The discrete-time time-optimal control seems to work quite well, if a relatively long sampling interval can be tolerated. If a higher sampling time is required, a sub-optimal control must be used. One possibility is to use deadbeat control, which in this case would allow 7 sample periods to track the command. The deadbeat algorithm is not presently in the INTERAC package, but it could be easily added. The second possibility is to place the desired eigenvalues at some place other than the origin in the z-plane. Since there are an infinite number of possible eigenvalues to use, the INTERAC package is more desirable than the batch mode program for quickly trying the most promising choices of eigenvalues.

V. Conclusions and Recommendations

Conclusions

The INTERAC package meets all of the specifications given in Chapter 1. There is still a little room for improvement in the I-O versatility, but what is now presented is a vast improvement over existing programs. In a current investigation, Lt. Stanley J. Larimer is incorporating many of these user oriented ideas into a classical control design package.

The use of discrete-time time-optimal control systems for total aircraft control is not generally possible due to the long sampling interval required to keep the magnitude of the control within practical limits. There are applications, with low frequency disturbances and commands, where the time-optimal control could be used, such as cruise control and possibly automatic approach control. The total aircraft control system will probably require a sub-optimal control law.

The theory behind the FORTRAC and INTERAC packages does not provide a cut and dry control system design process. There are too many possible parameter changes in the multi-input, multi-output system for a one best answer to be achieved in one iteration. The INTERAC package provides a basis for quickly investigating the various parameter changes.

Recommendations

(1) A graphical simulation capability should be added to the INTERAC package. The magnitude of the feedback gains does give a general idea as to the responce of the closed loop system. But, as

shown for the sampling interval of .6 and.7 seconds in Table VIII, there can be a drastic change in the total system response without large changes in the feedback gains.

(2) The DEADBEAT subroutine for achieving the sub-optimal control law should be added to the package. This subroutine is in the FORTRAC package, but it is not called by the master program furnished.

(3) The investigation of the discrete-time fly-by-wire control system for the C-141 should be continued with basic flight test data from Lockheed. The heavy transport type aircraft does not normally require high frequency control and response, and thus it would be ideal for further investigating the time-optimal control laws.

Bibliography

1. ROOTL - User's Manual for a Digital Computer Program to Calculate and Plot the Root Locus, Unpublished user's guide to computer program. Ohio: Air Force Institute of Technology, January 1976.

2. FREQR - Frequency Response Program, Unpublished user's guide to computer program. Ohio: Air Force Institute of Technology, November 1975.

3. PARTL - Heaviside Partial Fraction Expansion and Time Response Program, Unpublished user's guide to computer program. Ohio : Air Force Institute of Technology, December 1974.

4. OPTCON - Solution of Linear Quadratic Optimal Control Problem, Unpublished user's guide to computer program. Ohio: Air Force Institute of Technology, January 1976.

5. MIMIC Version 3.4, Unpublished user's guide to computer program. Ohio: Air Force Institute of Technology, August 1974.

6. Porter, B. FORTRAC: A Software Package for the Design of Multivariable Digital Control Systems, Unpublished report. Salford, England: University of Salford, April 1977.

7. Porter, B. and Bradshaw, A. Unpublished set of 10 papers on discrete-time control. Salford, England: University of Salford.

8. Aplevich, J.D. "Direct Computation of Canonical Forms for Linear Systems by Elementry Matrix Operations." IEEE Transactions on Automatic Control, AC-19: pp 124-126 (April 1974)

9. Nikolai, Paul J. Solution of the Eigenproblem on the CDC-6600/ Cyber-74 Processors. AFFDL-TM-76-131-FBR. Ohio: Air Force Flight Dynamics Lab, January 1977.

10. Control Data Corporation, FORTRAN Extended Version 4 Reference Manual. California: Control Data Corporation, 1975.

11. Control Data Corporation, SCOPE 3.4 Reference Manual. California: Control Data Corporation, 1975.

12. Honeywell, Inc. Military Transport (C-141) Fly-by-Wire Program, Vol I. AFFDL-TR-74-52. Minnesota: Honeywell, Inc., April 1974.

13. EE 6.41 Automatic Control Lab, Laboratory Experiment #3, Unpublished laboratory problem and data definition. Ohio: Air Force Institute of Technology, Sprint Qtr 1977.

14. Malcolm, Tobie. New Short Period Handling Qualities Criterion for Fighter Aircraft (C^* Control Criterion). Washington: The Boeing Company, September 1965.

Appendix A

Listing of Specialized INTERAC Software

The following pages are a listion of the FORTRAN Extended source code for the specialized INTERAC software.

```
      PROGRAM INTERAC(INPUT=100B/80,OUTPUT=1000B,TAPE5=INPUT,             INTERAC    2
     1                TAPE6=OUTPUT,TAPE8=100B,TAPE7=100B,ANSWER=100B,     INTERAC    3
     2                TAPE1=100B,TAPE2=100B,TAPE3=100B,DATA=500B,         INTERAC    4
     3                TAPE99=ANSWER,TAPE55=DATA,TAPE65=500B)              INTERAC    5
      COMMON/LISTING/NL,LIST(75,7)                                        INTERAC    6
      COMMON/LISTOUT/N,LISTO(75)                                          INTERAC    7
C                                                                         INTERAC    8
C     WHEN CHANGING DIMENSIONS OF THE ARRAY LIST THE ROW DIMENSION OF     INTERAC    9
C     LISTO AND THE DIMENSION STATEMENTS FOR LIST IN SUBROUTINES          INTERAC   10
C     FORTRAC,COMND, OPEN, AND STRING MUST ALSO BE CHANGED ACCORDINGLY.   INTERAC   11
C                                                                         INTERAC   12
      COMMON/RVAR/NTOT,TOT(3500)                                          INTERAC   13
      COMMON/IVAR/NITOT,ITOT(25)                                          INTERAC   14
      COMMON/CONTROL/DSP,BATCH                                            INTERAC   15
      COMMON/TITLE/ITITLE(5)                                              INTERAC   15
      COMMON/FORMS/NDREP,NDFW,NONSD,NOREP,NOFW,NONSO                      INTERAC   17
      DATA (LIST(  1,I),I=1,7),LISTO(  1)/10HENTER     E,3HCMD,           INTERAC   18
     1                                    1, 0, 0, 0, 0,1/                INTERAC   19
      DATA (LIST(  2,I),I=1,7),LISTO(  2)/10HSTOP      O,3HCMD,           INTERAC   20
     1                                    2, 0, 0, 0,-1,1/                INTERAC   21
      DATA (LIST(  3,I),I=1,7),LISTO(  3)/10HAMATRIX   G,3HREL,           INTERAC   22
     1                                    1, 1, 1,20,20,4/                INTERAC   23
      DATA (LIST(  4,I),I=1,7),LISTO(  4)/10HBMATRIX   G,3HREL,           INTERAC   24
     1                                  401, 1, 1,20,10,4/                INTERAC   25
      DATA (LIST(  5,I),I=1,7),LISTO(  5)/10HEND       C,3HCMD,           INTERAC   26
     1                                    3, 0, 0, 0,-1,1/                INTERAC   27
      DATA (LIST(  6,I),I=1,7),LISTO(  6)/10HPRINT     E,3HCMD,           INTERAC   28
     1                                    4, 0, 0, 0, 0,1/                INTERAC   29
      DATA (LIST(  7,I),I=1,7),LISTO(  7)/10HDFORMAT   G,3HCMD,           INTERAC   30
     1                                    5, 0, 0, 0,-1,1/                INTERAC   31
      DATA (LIST(  8,I),I=1,7),LISTO(  8)/10HCHANGE    F,3HCMD,           INTERAC   32
     1                                    6, 0, 0, 0, 0,1/                INTERAC   33
      DATA (LIST(  9,I),I=1,7),LISTO(  9)/10HDISPLAY   G,3HCMD,           INTERAC   34
     1                                    7, 0, 0, 0, 0,1/                INTERAC   35
      DATA (LIST( 10,I),I=1,7),LISTO( 10)/10HOUTPUT    F,3HCMD,           INTERAC   36
     1                                     8, 0, 0, 0, 0,1/                   INTERAC    37
      DATA (LIST( 11,I),I=1,7),LISTO( 11)/10HDELETE    F,3HCMD,               INTERAC    38
     1                                     9, 0, 0, 0, 0,1/                   INTERAC    39
      DATA (LIST( 12,I),I=1,7),LISTO( 12)/10HOPTIONS   G,3HCMD,               INTERAC    40
     1                                    10, 0, 0, 0,-1,1/                   INTERAC    41
      DATA (LIST( 13,I),I=1,7),LISTO( 13)/10HVARIABLESI,3HCMD,                INTERAC    42
     1                                    11, 0, 0, 0,-1,1/                   INTERAC    43
      DATA (LIST( 14,I),I=1,7),LISTO( 14)/10HSAVE      D,3HCMD,               INTERAC    44
     1                                    12, 0, 0, 0,-1,1/                   INTERAC    45
      DATA (LIST( 15,I),I=1,7),LISTO( 15)/10HRESTART   G,3HCMD,               INTERAC    46
     1                                    13, 0, 0, 0, 0,1/                   INTERAC    47
      DATA (LIST( 16,I),I=1,7),LISTO( 16)/10HREWIND    F,3HCMD,               INTERAC    48
     1                                    14, 0, 0, 0,-2,1/                   INTERAC    49
      DATA (LIST( 17,I),I=1,7),LISTO( 17)/10HRUN       C,3HCMD,               INTERAC    50
     1                                    15, 0, 0, 0, 1,1/                   INTERAC    51
      DATA (LIST( 18,I),I=1,7),LISTO( 18)/10HDEIGEN    F,3HREL,               INTERAC    52
     1                                  1741, 1, 1,20, 2,2/                   INTERAC    53
      DATA (LIST( 19,I),I=1,7),LISTO( 19)/10HDIMENSIONI,3HINT,                INTERAC    54
     1                                     1, 1, 4, 1, 1,4/                   INTERAC    55
      DATA (LIST( 20,I),I=1,7),LISTO( 20)/10HFORTRAC   G,3HCMD,               INTERAC    56
     1                                    16, 0, 0, 1,-2,1/                   INTERAC    57
      DATA (LIST( 21,I),I=1,7),LISTO( 21)/10HSAMPLE    F,3HCMD,               INTERAC    58
     1                                    17, 0, 0, 2,-2,1/                   INTERAC    59
      DATA (LIST( 22,I),I=1,7),LISTO( 22)/10HAUGMAT    F,3HCMD,               INTERAC    60
     1                                    18, 0, 0, 3,-2,1/                   INTERAC    61
      DATA (LIST( 23,I),I=1,7),LISTO( 23)/10HTRANSFORMI,3HCMD,                INTERAC    62
     1                                    19, 0, 0, 4,-2,1/                   INTERAC    63
      DATA (LIST( 24,I),I=1,7),LISTO( 24)/10HCONLAW    F,3HCMD,               INTERAC    64
     1                                    20, 0, 0, 5,-2,1/                   INTERAC    65
      DATA (LIST( 25,I),I=1,7),LISTO( 25)/10HCLOSELOOPI,3HCMD,                INTERAC    66
     1                                    21, 0, 0, 6,-2,1/                   INTERAC    67
      DATA (LIST( 26,I),I=1,7),LISTO( 26)/10HCOFMATRIXI,3HOUT,                INTERAC    68
     1                                     0, 0, 0, 0, 0,1/                   INTERAC    69
      DATA (LIST( 27,I),I=1,7),LISTO( 27)/10HINTEGRATEI,3HINT,                INTERAC    70
     1                                     5, 1, 1, 1,10,1/                   INTERAC    71
```

```
      DATA (LIST( 28,I),I=1,7),LISTO( 28)/10HOFORMAT  G,3HCMD,              INTERAC    72
     1                               22, 0, 0, 0,-1,1/                      INTERAC    73
      DATA (LIST( 29,I),I=1,7),LISTO( 29)/10HSETDIMEN H,3HCMD,              INTERAC    74
     1                               23, 0, 0, 0,-1,1/                      INTERAC    75
      DATA (LIST( 30,I),I=1,7),LISTO( 30)/10HRMATRIX  G,3HREL,              INTERAC    76
     1                              601, 1, 1,20, 5,4/                      INTERAC    77
      DATA (LIST( 31,I),I=1,7),LISTO( 31)/10HCMATRIX  G,3HREL,              INTERAC    78
     1                              701, 1, 1,10,20,4/                      INTERAC    79
      DATA (LIST( 32,I),I=1,7),LISTO( 32)/10HFMATRIX  G,3HREL,              INTERAC    80
     1                              901, 1, 1,20,20,3/                      INTERAC    81
      DATA (LIST( 33,I),I=1,7),LISTO( 33)/10HGMATRIX  G,3HREL,              INTERAC    82
     1                             1301, 1, 1,20,10,3/                      INTERAC    83
      DATA (LIST( 34,I),I=1,7),LISTO( 34)/10HDEMATRIX H,3HREL,              INTERAC    84
     1                             1501, 1, 1,20, 5,1/                      INTERAC    85
      DATA (LIST( 35,I),I=1,7),LISTO( 35)/10HDRMATRIX H,3HREL,              INTERAC    86
     1                             1601, 1, 1,20, 5,1/                      INTERAC    87
      DATA (LIST( 36,I),I=1,7),LISTO( 36)/10HAFMATRIX H,3HREL,              INTERAC    88
     1                              901, 1, 1,20,20,4/                      INTERAC    89
      DATA (LIST( 37,I),I=1,7),LISTO( 37)/10HAGMATRIX H,3HREL,              INTERAC    90
     1                             1301, 1, 1,20,10,4/                      INTERAC    91
      DATA (LIST( 38,I),I=1,7),LISTO( 38)/10HAEMATRIX H,3HREL,              INTERAC    92
     1                             1501, 1, 1,20, 5,4/                      INTERAC    93
      DATA (LIST( 39,I),I=1,7),LISTO( 39)/10HARMATRIX H,3HREL,              INTERAC    94
     1                             1601, 1, 1,20, 5,4/                      INTERAC    95
      DATA (LIST( 40,I),I=1,7),LISTO( 40)/10HACMATRIX H,3HREL,              INTERAC    96
     1                              701, 1, 1,10,20,4/                      INTERAC    97
      DATA (LIST( 41,I),I=1,7),LISTO( 41)/10HCEIGEN   F,3HREL,              INTERAC    98
     1                             1701, 1, 1,20, 2,1/                      INTERAC    99
      DATA (LIST( 42,I),I=1,7),LISTO( 42)/10HCLEIGEN  G,3HREL,              INTERAC   100
     1                             1781, 1, 1,20, 2,3/                      INTERAC   101
      DATA (LIST( 43,I),I=1,7),LISTO( 43)/10HBFMATRIX H,3HREL,              INTERAC   102
     1                             1821, 1, 1,20,20,1/                      INTERAC   103
      DATA (LIST( 44,I),I=1,7),LISTO( 44)/10HBGMATRIX H,3HREL,              INTERAC   104
     1                             2221, 1, 1,20,10,1/                      INTERAC   105
      DATA (LIST( 45,I),I=1,7),LISTO( 45)/10HTINVERSE H,3HREL,              INTERAC   106
     1                                   2421, 1, 1,20,20,1/              INTERAC   107
      DATA (LIST( 46,I),I=1,7),LISTO( 46)/10HKMATRIX   G,3HREL,           INTERAC   108
     1                                   2921, 1, 1,10,20,3/              INTERAC   109
      DATA (LIST( 47,I),I=1,7),LISTO( 47)/10HDESEIGEN H,3HREL,            INTERAC   110
     1                                   3422, 1, 1,20, 2,4/              INTERAC   111
      DATA (LIST( 48,I),I=1,7),LISTO( 48)/10HCLMATRIX H,3HREL,            INTERAC   112
     1                                   3021, 1, 1,20,20,1/              INTERAC   113
      DATA (LIST( 49,I),I=1,7),LISTO( 49)/10HTSAMPLE  G,3HREL,            INTERAC   114
     1                                   3421, 1, 1, 1, 1,3/              INTERAC   115
      DATA (LIST( 50,I),I=1,7),LISTO( 50)/10HSTATES   F,3HINT,            INTERAC   116
     1                                      1, 1, 1, 1, 1,1/              INTERAC   117
      DATA (LIST( 51,I),I=1,7),LISTO( 51)/10HCONTROLS H,3HINT,            INTERAC   118
     1                                      2, 1, 1, 1, 1,1/              INTERAC   119
      DATA (LIST( 52,I),I=1,7),LISTO( 52)/10HCOMMANDS H,3HINT,            INTERAC   120
     1                                      3, 1, 1, 1, 1,1/              INTERAC   121
      DATA (LIST( 53,I),I=1,7),LISTO( 53)/10HDISTURBS H,3HINT,            INTERAC   122
     1                                      4, 1, 1, 1, 1,1/              INTERAC   123
      DATA (LIST( 54,I),I=1,7),LISTO( 54)/10HMODAL    E,3HOUT,            INTERAC   124
     1                                      0, 0, 0, 0, 0,4/              INTERAC   125
      DATA (LIST( 55,I),I=1,7),LISTO( 55)/10HINVMODAL H,3HOUT,            INTERAC   126
     1                                      0, 0, 0, 0, 0,1/              INTERAC   127
      DATA (LIST( 56,I),I=1,7),LISTO( 56)/10HCINDICES H,3HOUT,            INTERAC   128
     1                                      0, 0, 0, 0, 0,2/              INTERAC   129
      DATA (LIST( 57,I),I=1,7),LISTO( 57)/10HUREFORM  G,3HOUT,            INTERAC   130
     1                                      0, 0, 0, 0, 0,1/              INTERAC   131
      DATA (LIST( 58,I),I=1,7),LISTO( 58)/10HAPLFORM  G,3HOUT,            INTERAC   132
     1                                      0, 0, 0, 0, 0,1/              INTERAC   133
      DATA (LIST( 59,I),I=1,7),LISTO( 59)/10HBRNFORM  G,3HOUT,            INTERAC   134
     1                                      0, 0, 0, 0, 0,2/              INTERAC   135
      DATA NL,N,NTOT,NITOT/59,59,3500,25/                                 INTERAC   136
      DATA NOREP,NOFW,NONSO/5,13,5/                                       INTERAC   137
      DATA NOREP,NOFW,NONSO/10,12,4/                                      INTERAC   138
      DATA ITITLE/5*1H /                                                  INTERAC   139
      CALL CCMND                                                          INTERAC   140
      END                                                                 INTERAC   141
      SUBROUTINE COMND
      COMMON/LISTING/NL,LIST(75,7)
      COMMON/LISTOUT/NO,LISTO(1)
      COMMON/RVAR/NTOT,TOT(1)
      COMMON/IVAR/NITOT,ITOT(1)
      COMMON/CONTROL/DSP,BATCH
      COMMON/FORMS/NDREP,NDFW,NDNSD,NOREP,NOFW,NONSD
      LOGICAL BATCH
      INTEGER DSP,CMD(12),CMST,CMED
      CALL CONNEC(5LINPUT)
      CALL CONNEC(5LTAPE8)
      CALL DATE(TODAY)
      CALL TIME(CLOCK)
      PRINT(8,*)"  IF YOU ARE AT A TERMINAL TYPE TTY  >"
      READ 5,JCN,KCN
    5 FORMAT(R3,1X,R1)
      IF(JCN.EQ.3RTTY) GO TO 10
      DSP=6
      BATCH=.TRUE.
      GO TO 12
   10 DSP=8
      BATCH=.FALSE.
      CALL DISCON(6LOUTPUT)
      REWIND 6
   12 PRINT(6,15)TODAY,CLOCK
   15 FORMAT("1",10X,"START      DATE ",A10,5X,"TIME ",A10,////)
      IF(KCN.NE.1RS) GO TO 5000
   20 PRINT(8,*)
      PRINT(8,*)
      PRINT(8,*)"  COMMAND >>"
      NUM=12
      CALL READCOM(CMD,NUM)
      IF(CMD(1).EQ.0) GO TO 90
      CMED=0
   25 CMST=CMED+2
```

```
      K=0                                                                COMND
      DO 30 I=CMST,NUM                                                   COMND
      IF(CMD(I).GE.0) GO TO 35                                           COMND
      K=K+1                                                              COMND
   30 CONTINUE                                                           COMND
   35 CMST=CMST-1                                                        COMND
      IF(K.EQ.0) GO TO 50                                                COMND
      IF(LIST(CMD(CMST),2).NE.3HCMD) GO TO 45                            COMND
      IF(LIST(CMD(CMST),7).LT.0 .OR. LIST(CMD(CMST),7).GT.K) GO TO 45    COMND
      CALL STRING(LIST(CMD(CMST),3),CMD(CMST+1),K),RETURNS(20)           COMND
      GO TO 40                                                           COMND
   40 CMED=CMST+K                                                        COMND
      IF(CMED.EQ.NUM) GO TO 20                                           COMND
      GO TO 25                                                           COMND
   45 PRINT(8,46)AND(LIST(CMD(CMST),1),77B),LIST(CMD(CMST),1),           COMND
     1 AND(LIST(-CMD(CMST+K),1),77B),LIST(-CMD(CMST+K),1)                COMND
      IF(BATCH) STOP                                                     COMND
      GO TO 40                                                           COMND
   46 FORMAT(/,5X,"COMMAND STRING",3X,A=,".......",A=,3X,                COMND
     1  "ILLEGAL AND IGNORED.",/)                                        COMND
   50 JCN=CMD(CMST)                                                      COMND
      IF(LIST(JCN,2).EQ.3HOUT) GO TO 60                                  COMND
      IF(LIST(JCN,2).EQ.3HREL .OR. LIST(JCN,2).EQ.3HINT) GO TO 75        COMND
      IF(LIST(JCN,7).EQ.-3 .OR. LIST(JCN,7).GT.0) GO TO 60               COMND
      GO TO ( 100, 200, 300, 400, 500, 600, 700, 800, 900,1000,          COMND
     1       1100,1200,1300,1400,3000,1600,1700,1800,1900,2000,          COMND
                                                                         COMND
     2       2100,2200,2300,3000,3000,3000,3000,3000,3000,3000)          COMND
     3 LIST(JCN,3)                                                       COMND
   60 PRINT(8,65)AND(LIST(JCN,1),77B),LIST(JCN,1)                        COMND
      GO TO 40                                                           COMND
   65 FORMAT(/10X,A=,3X,"ILLEGAL AND IGNORED",/)                         COMND
   75 CALL OPEN(JCN),RETURNS(20)                                         COMND
      GO TO 40                                                           COMND
   90 IF(BATCH) STOP                                                     COMND
```

```
      GO TO 20                                                          COMND      72
C*****ENTER**                                                           COMND      73
  100 PRINT(8,150)                                                      COMND      74
      NC=1                                                              COMND      75
      CALL READCOM(ICOM,NC)                                             COMND      76
      IF(ICOM.EQ.0) GO TO 90                                            COMND      77
      IF(LIST(ICOM,1).EQ.10HEND       C) GO TO 40                       COMND      78
      IF(LIST(ICOM,2).EQ.3HCMD .OR. LIST(ICOM,2).EQ.3HOUT) GO TO 120    COMND      79
      CALL OPEN(ICOM),RETURNS(20)                                       COMND      80
      GO TO 100                                                         COMND      81
  120 PRINT(8,65)AND(LIST(ICOM,1),778),LIST(ICOM,1)                     COMND      82
      GO TO 100                                                         COMND      83
  150 FORMAT(/10X,"ENTER VARIABLE NAME OR END  >>")                     COMND      84
C*****STOP**    **END**                                                 COMND      85
  200 PRINT(8,250)                                                      COMND      86
  250 FORMAT(/,5X,"INTERAC TERMINATED, ANSWER IS DISCONNECTED AND CONTAI COMND      87
     1NS",/," YOUR PRINTED DATA.  FILE DATA HAS THE CURRENT CONTENTS",  COMND      88
     2/," OF ALL VARIABLES;  THEY CAN BE RELOADED AT SOME FUTURE TIME BY COMND      89
     3",/," THE COMMAND RESTART.")                                      COMND      90
  300 CALL DATE(TODAY)                                                  COMND      91
      CALL TIME(CLOCK)                                                  COMND      92
      PRINT(55,340)TODAY,CLOCK                                          COMND      93
      PRINT(8,345)TODAY,CLOCK                                           COMND      94
  340 FORMAT(2A10)                                                      COMND      95
  345 FORMAT(5X,"DATA SAVED AT ",2A10)                                  COMND      96
      PRINT(55,*)(TOT(I),I=1,NTOT)                                      COMND      97
      PRINT(55,*)(ITOT(I),I=1,NITOT)                                    COMND      98
      PRINT(55,*)((LIST(I,4),LIST(I,5)),I=1,NL)                         COMND      99
      PRINT(55,*)(LISTO(I),I=1,NO)                                      COMND     100
      CALL RETURN(5LTAPE7)                                              COMND     101
      CALL RETURN(5LTAPE8)                                              COMND     102
      CALL RETURN(5LTAPE55)                                             COMND     103
      PRINT(8,350)TODAY,CLOCK                                           COMND     104
  350 FORMAT("2",10X,"STOP     DATE ",A10,5X,"TIME ",A10)               COMND     105
      CALL FORMOUT                                                      COMND     106
```

```
C*****PRINT**
  400 PRINT(8,150)
      NC=1
      CALL RFADCOM(ICOM,NC)
      IF(ICOM.EQ.0) GO TO 90
      IF(LIST(ICOM,1).EQ.10HEND        C) GO TO 40
      IF(LIST(ICOM,2).EQ.3HCMD .OR. LIST(ICOM,2).EQ.3HOUT) GO TO 420
      NP=LIST(ICOM,4)
      MP=LIST(ICOM,5)
      NDP=LIST(ICOM,6)
      IF(LIST(ICOM,2).EQ.3HINT) GO TO 410
      CALL PPINTR(0,LIST(ICOM,1),TOT(LIST(ICOM,3)),NP,MP,NDP)
      GO TO 400
  410 CALL PPINTI(0,LIST(ICOM,1),ITOT(LIST(ICOM,3)),NP,MP,NDP)
      GO TO 400
  420 PRINT(8,65)AND(LIST(ICOM,1),77B),LIST(ICOM,1)
      GO TO 400
C*****DFORMAT**
  500 PRINT(8,550)
      CALL RFADNUM(10H         A,NC,1,1,1,.FALSE.),RETURNS(20)
      IF(NC.LT.1 .OR. NC.GT.14) GO TO 500
      NDREP=66/(NC+7)
      NDFW=NC+7
      NDNSD=NC-1
      GO TO 40
  550 FORMAT(5X,"ENTER NUMBER OF SIGNIFICANT DIGITS  0 < N < 15."/)
C*****CHANGE**
  600 PRINT(8,150)
      NC=1
      CALL READCOM(ICOM,NC)
      IF(ICOM.EQ.0) GO TO 90
      IF(LIST(ICOM,1).EQ.10HEND        C) GO TO 40
      IF(LIST(ICOM,2).EQ.3HCMD) GO TO 620
      CALL RFOPEN(ICOM),RETURNS(20)
      GO TO 600
  620 PRINT(8,65)AND(LIST(ICOM,1),77B),LIST(ICOM,1)
      GO TO 600
C*****DISPLAY**
  700 PRINT(8,*)"  VARIABLES TO BE DISPLAYED AT TERMINAL"
      DO 720 J=1,NL
      IF(LISTO(J).NE.2 .AND. LISTO(J).NE.3) GO TO 720
      PRINT(8,750)AND(LIST(J,1),77B),LIST(J,1)
  720 CONTINUE
      GO TO 40
  750 FORMAT(5X,A=)
C*****OUTPUT**
  800 PRINT(8,*)"  VARIABLES TO BE PRINTED TO ANSWER"
      DO 820 J=1,NL
      IF(LISTO(J).NE.3 .AND. LISTO(J).NE.4) GO TO 820
      PRINT(8,850)AND(LIST(J,1),77B),LIST(J,1)
  820 CONTINUE
      GO TO 40
  850 FORMAT(5X,A=)
C*****DELETE**
  900 PRINT(8,*)"  ALL VARIABLES HAVE BEEN REMOVED FROM THE DISPLAY AND
     1OUTPUT LISTS."
      DO 920 J=1,NL
      LISTO(J)=1
  920 CONTINUE
      GO TO 40
C*****OPTIONS**
 1000 PRINT(8,*)"  VALID COMMANDS"
      DO 1020 J=1,NL
      IF(LIST(J,2).NE.3HCMD) GO TO 1020
      IF(LIST(J,7)) 1005,1010,1015
 1005 PRINT(8,1050)AND(LIST(J,1),77B),LIST(J,1)
      GO TO 1020
 1010 PRINT(8,1060)AND(LIST(J,1),77B),LIST(J,1)
      GO TO 1020
 1015 PRINT(8,1070)AND(LIST(J,1),77B),LIST(J,1)
 1020 CONTINUE                                                                 COMND
      GO TO 40                                                                 COMND
 1050 FORMAT(5X,A=)                                                            COMND
 1060 FORMAT(5X,A=,T18,"{,PARAMETER LIST}")                                    COMND
 1070 FORMAT(5X,A=,T18,",PARAMETER(S)")                                        COMND
C*****VARIABLES**                                                              COMND
 1100 PRINT(8,*)"  VALID VARIABLES          MAX DIMENSION ALLOWED"             COMND
      DO 1120 J=1,NL                                                           COMND
      IF(LIST(J,2).EQ.3HCMD) GO TO 1120                                        COMND
      ISIM=1H                                                                  COMND
      IF(LIST(J,2).EQ.3HOUT) ISIM=1H*                                          COMND
      PRINT(8,1150)ISIM,AND(LIST(J,1),77B),LIST(J,1),LIST(J,6),LIST(J,7) COMND
 1120 CONTINUE                                                                 COMND
      PRINT(8,*)                                                               COMND
      PRINT(8,*)"  * -- TEMPORARY VARIABLE PRINTED ONLY DURING PROGRAM E COMND
     ^XECUTION."                                                               COMND
      GO TO 40                                                                 COMND
 1150 FORMAT(5X,A2,A=,T33,"(",I3,",",I3,")")                                   COMND
C*****SAVE**                                                                   COMND
 1200 CALL DATE(TODAY)                                                         COMND
      CALL TIME(CLOCK)                                                         COMND
      PRINT(55,340)TODAY,CLOCK                                                 COMND
      PRINT(8,345)TODAY,CLOCK                                                  COMND
      PRINT(55,*)(TOT(I),I=1,NTOT)                                             COMND
      PRINT(55,*)(ITOT(I),I=1,NITOT)                                           COMND
      PRINT(55,*)((LIST(I,4),LIST(I,5)),I=1,NL)                                COMND
      PRINT(55,*)(LISTO(I),I=1,NO)                                             COMND
      GO TO 40                                                                 COMND
C*****RESTART**                                                                COMND
 1300 REWIND 65                                                                COMND
      PRINT(65,*)(TOT(I),I=1,NTOT)                                             COMND
      PRINT(65,*)(ITOT(I),I=1,NITOT)                                           COMND
      PRINT(65,*)((LIST(I,4),LIST(I,5)),I=1,NL)                                COMND
      PRINT(65,*)(LISTO(I),I=1,NO)                                             COMND
      READ(55,340)TODAY,CLOCK                                                  COMND
      IF(EOF(6LTAPE55).NE.0.0) GO TO 1350
      READ(55,*)(TOT(I),I=1,NTOT)
      IF(EOF(6LTAPE55).NE.0.0) GO TO 1350
      READ(55,*)(ITOT(I),I=1,NITOT)
      IF(EOF(6LTAPE55).NE.0.0) GO TO 1350
      READ(55,*)((LIST(I,4),LIST(I,5)),I=1,NL)
      IF(EOF(6LTAPE55).NE.0.0) GO TO 1350
      READ(55,*)(LISTO(I),I=1,NO)
      IF(EOF(6LTAPE55).NE.0.0) GO TO 1350
      PRINT(8,1340)TODAY,CLOCK
 1340 FORMAT(/,5X,"DATA SAVED AT ",2A10," RELOADED.")
      GO TO 40
 1350 PRINT(8,*)"  EOF ENCOUNTERED ON RESTART - REWIND IS PROBABLY NECES
     1SARY"
      REWIND 65
      READ(65,*)(TOT(I),I=1,NTOT)
      READ(65,*)(ITOT(I),I=1,NITOT)
      READ(65,*)((LIST(I,4),LIST(I,5)),I=1,NL)
      READ(65,*)(LISTO(I),I=1,NO)
      GO TO 40
C*****REWIND**
 1400 REWIND 55
      GO TO 40
C*****FORTRAC**
 1600 CALL FORTRAC(1)
      GO TO 40
C*****SAMPLE**
 1700 CALL FORTRAC(2)
      GO TO 40
C*****AUGMAT**
 1800 CALL FORTRAC(3)
      GO TO 40
C*****TRANSFORM**
 1900 CALL FORTRAC(4)
      GO TO 40
```

```
C*****CONTROL**
 2000 CALL FORTRAC(5)
      GO TO 40
C*****CLOSELCOF**
 2100 CALL FORTRAC(6)
      GO TO 40
C*****OFORMAT**
 2200 PRINT(8,550)
      CALL READNUM(10H          A,NC,1,1,1,.FALSE.),RETURNS(20)
      IF(NC.LT.1 .OR. NC.GT.14) GO TO 2200
      NOREP=126/(NC+7)
      NOFW=NC+7
      NONSD=NC-1
      GO TO 40
C*****SETDIMEN**
 2300 CALL FORTRAC(100)
      GO TO 40
C     **NOT USED YET**
 3000 PRINT(8,*)"  COMMAND NOT AVAILABLE"
      GO TO 40
C*****INTRODUCTION**
 5000 PRINT(8,5001)
 5001 FORMAT(/,10X,"WELCOME TO INTERAC:  A SOFTWARE PACKAGE FOR",/,
     1  5X,"DIRECT DIGITAL CONTROL SYSTEM DESIGN FOR",/,
     2  5X,"LINEAR MULTI-INPUT, MULTI-OUTPUT PLANTS.  FOR A",/,
     3  5X,"STEP BY STEP PROCEEDURE IN USING THIS PACKAGE",/,
     4  5X,"TYPE  FORTRAC.")
      GO TO 20
      END
      SUBROUTINE STRING(N,CMD,K),RETURNS(ABORT)
      COMMON/LISTING/NL,LIST(75,7)
      COMMON/LISTOUT/NO,LISTO(1)
      COMMON/RVAR/NTOT,TOT(1)
      COMMON/IVAR/NITOT,ITOT(1)
      COMMON/TITLE/ITITLE(5)
      INTEGER CMD(K)
      DO 2 I=1,K
    2 CMD(I)=-CMD(I)
      GO TO ( 10,500,500, 40,500, 60, 70 ,80, 90,500,
     1        500,500,130,500,150,500,500,500,500,500,
     2        500,500,500,500,500,500,500,500,500,500) N
    5 RETURN ABORT
C*****ENTER**
   10 DO 18 I=1,K
      IF(LIST(CMD(I),2).EQ.3HCMD .OR. LIST(CMD(I),2).EQ.3HOUT) GO TO 15
      CALL OPEN(CMD(I)),RETURNS(5)
      GO TO 18
   15 PRINT(8,16)AND(LIST(CMD(I),1),77B),LIST(CMD(I),1)
   16 FORMAT(/,5X,"ENTER,",A=,"  ILLEGAL AND IGNORED")
   18 CONTINUE
      RETURN
C*****PRINT**
   40 DO 48 I=1,K
      IF(LIST(CMD(I),2).EQ.3HCMD .OR. LIST(CMD(I),2).EQ.3HOUT) GO TO 45
      NP=LIST(CMD(I),4)
      MP=LIST(CMD(I),5)
      NDP=LIST(CMD(I),6)
      IF(LIST(CMD(I),2).EQ.3HINT) GO TO 44
      CALL PRINTR(0,LIST(CMD(I),1),TOT(LIST(CMD(I),3)),NP,MP,NDP)
      GO TO 48
   44 CALL PRINTI(0,LIST(CMD(I),1),ITOT(LIST(CMD(I),3)),NP,MP,NDP)
      GO TO 48
   45 PRINT(8,46)AND(LIST(CMD(I),1),77B),LIST(CMD(I),1)
   46 FORMAT(/,5X,"PRINT,",A=,"  ILLEGAL AND IGNORED")
   48 CONTINUE                                                              STRING
      RETURN                                                                STRING
C*****CHANGE**                                                              STRING
   60 DO 68 I=1,K                                                           STRING
      IF(LIST(CMD(I),2).EQ.3HCMD .OR. LIST(CMD(I),2).EQ.3HOUT) GO TO 65     STRING
      CALL REOPEN(CMD(I)),RETURNS(5)                                        STRING
      GO TO 68                                                              STRING
   65 PRINT(8,66)AND(LIST(CMD(I),1),77B),LIST(CMD(I),1)                     STRING
   66 FORMAT(/,5X,"CHANGE,",A=,"  ILLEGAL AND IGNORED")                     STRING
   68 CONTINUE                                                              STRING
      RETURN                                                                STRING
C*****DISPLAY**                                                             STRING
   70 DO 78 I=1,K                                                           STRING
      IF(LIST(CMD(I),2).EQ.3HCMD) GO TO 75                                  STRING
      ITEMP=LISTO(CMD(I))                                                   STRING
      LISTO(CMD(I))=2                                                       STRING
      IF(ITEMP.EQ.4) LISTO(CMD(I))=3                                        STRING
      GO TO 78                                                              STRING
   75 PRINT(8,76)AND(LIST(CMD(I),1),77B),LIST(CMD(I),1)                     STRING
   76 FORMAT(/,5X,"DISPLAY,",A=,"  ILLEGAL AND IGNORED")                    STRING
   78 CONTINUE                                                              STRING
      RETURN                                                                STRING
C*****OUTPUT**                                                              STRING
   80 DO 88 I=1,K                                                           STRING
      IF(LIST(CMD(I),2).EQ.3HCMD) GO TO 85                                  STRING
      ITEMP=LISTO(CMD(I))                                                   STRING
      LISTO(CMD(I))=4                                                       STRING
      IF(ITEMP.EQ.2) LISTO(CMD(I))=3                                        STRING
      GO TO 88                                                              STRING
   85 PRINT(8,86)AND(LIST(CMD(I),1),77B),LIST(CMD(I),1)                     STRING
   86 FORMAT(/,5X,"OUTPUT,",A=,"  ILLEGAL AND IGNORED")                     STRING
   88 CONTINUE                                                              STRING
      RETURN                                                                STRING
C*****DELETE**                                                              STRING
   90 DO 96 I=1,K                                                           STRING
```

```
      IF(LIST(CMD(I),2).EQ.3HCMD) GO TO 95
      LISTO(CMD(I))=1
      GO TO 98
   95 PRINT(8,96)AND(LIST(CMD(I),1),77B),LIST(CMD(I),1)
   96 FORMAT(/,5X,"DELETE,",A=,"  ILLEGAL AND IGNORED")
   98 CONTINUE
      RETURN
C*****RESTART**
  130 IF(K.GT.1) GO TO 135
      IF(LIST(CMD(1),1).NE.10HREWIND   F) GO TO 137
      REWIND 65
      PRINT(65,*)(TOT(I),I=1,NTOT)
      PRINT(65,*)(ITOT(I),I=1,NITOT)
      PRINT(65,*)((LIST(I,4),LIST(I,5)),I=1,NL)
      PRINT(65,*)(LISTO(I),I=1,NO)
      REWIND 55
      READ(55,131)TODAY,CLOCK
      IF(EOF(6LTAPE55).NE.0.0) GO TO 133
      READ(55,*)(TOT(I),I=1,NTOT)
      IF(EOF(6LTAPE55).NE.0.0) GO TO 133
      READ(55,*)(ITOT(I),I=1,NITOT)
      IF(EOF(6LTAPE55).NE.0.0) GO TO 133
      READ(55,*)((LIST(I,4),LIST(I,5)),I=1,NL)
      IF(EOF(6LTAPE55).NE.0.0) GO TO 133
      READ(55,*)(LISTO(I),I=1,NO)
      IF(EOF(6LTAPE55).NE.0.0) GO TO 133
      PRINT(8,132)TODAY,CLOCK
      RETURN
  131 FORMAT(2A10)
  132 FORMAT(/,5X,"DATA SAVED AT ",2A10," RELOADED.")
  133 PRINT(8,*)"  TAPE55 DOES NOT CONTAIN ANY DATA"
      REWIND 65
      READ(65,*)(TOT(I),I=1,NTOT)
      READ(65,*)(ITOT(I),I=1,NITOT)
      READ(65,*)((LIST(I,4),LIST(I,5)),I=1,NL)
      PEAD(65,*) (LISTO(I),I=1,NO)
      RETURN
  135 PRINT(8,136)AND(LIST(CMD(K),1),77B),LIST(CMD(K),1)
  136 FORMAT(/,5X,"COMMAND STRING  RESTART,...,",A=," ILLEGAL AND IGNORE
     1D")
      RETURN
  137 PRINT(8,138)AND(LIST(CMD,1),77B),LIST(CMD,1)
  138 FORMAT(/,5X,"RESTAPT,",A=,"  ILLEGAL AND IGNORED")
      RETURN
C*****RUN**
  150 IF(K.GT.1) GO TO 155
      IF(LIST(CMD(1),6).EQ.0 .OR. LIST(CMD(1),7).NE.-2) GO TO 157
      CALL FORTRAC(LIST(CMD(1),6)+10)
      RETURN
  155 PRINT(8,156)AND(LIST(CMD(K),1),77B),LIST(CMD(K),1)
  156 FORMAT(/,5X,"COMMAND STRING  RUN,...,",A=,"  ILLEGAL AND IGNORED")
      PETURN
  157 PPINT(8,158)AND(LIST(CMD(1),1),77B),LIST(CMD(1),1)
  158 FORMAT(/,5X,"RUN,",A=,"  ILLEGAL AND IGNORED")
      RETURN
  500 PRINT(8,*)" COMMAND STRING NOT AVAILABLE"
      RETURN
      END
      SUBROUTINE READNUM(NAME,MATRIX,N,M,ND,REAL),RETURNS(ABORT)             READNUM
      COMMON/CONTROL/DSP,BATCH                                              READNUM
      LOGICAL REAL,BATCH,LIST,END                                           READNUM
      REAL MATRIX(ND,M)                                                     READNUM
      INTEGER A(65),IN(73),COMMAND(6),AM(10),SLASH(2),                      READNUM
     ^AM1,GO,COMMA,REW,RET,WORD,POINTER,WORDST,DSP                          READNUM
      DATA A/66666666666666666666B,11111111111153555555B,                   READNUM
     ^11111111111131555553B,11111111111151555555B,11111111111131555553B,    READNUM
     ^11111111111151555555B,11111111111151555225B,11111111111151555555B,    READNUM
     ^11111111111151555555B,11111111111151555555B,11111111111151555555B,    READNUM
     ^11111111111151555555B,11111111111151555555B,11111111111131555553B,    READNUM
     ^11111111111151555555B,11111111111151555555B,11111111111151555555B,    READNUM
     ^11111111111151555555B,11111111111151555555B,11111111111151555553B,    READNUM
     ^11111111111151555555B,11111111111151555555B,11111111111151555555B,    READNUM
     ^11111111111151555555B,11111111111151555555B,11111111111151555555B,    READNUM
     ^11111111111151555555B,11111111111131555553B,                          READNUM
     ^10*11111111111131111112B,                                             READNUM
     ^11111111111113551552B,11111111111113551552B,11111111111113555552B,    READNUM
     ^11111111111142455554B,11111111111153555555B,11111111111153555555B,    READNUM
     ^11111111111135777773B,11111111111153555555B,11111111111112244441B,    READNUM
     ^11111111111112344441B,11111111111113555532B,11111111111153555555B,    READNUM
     ^11111111111153555555B,11111111111153555555B,11111111111153555555B,    READNUM
     ^11111111111153555555B,11111111111153555555B,11111111111153555555B,    READNUM
     ^11111111111153555555B,11111111111153555555B,11111111111153555555B,    READNUM
     ^11111111111153555555B,11111111111153555555B,11111111111153555555B,    READNUM
     ^11111111111153555555B,11111111111153555555B,11111111111153555555B/    READNUM
      DATA AM/00000000000000000077B,00000000000000007777B,                  READNUM
     ^00000000000000777777B,00000000000077777777B,                          READNUM
     ^00000000007777777777B,00000000777777777777B,                          READNUM
     ^00000077777777777777B,00007777777777777777B,                          READNUM
     ^00777777777777777777B,77777777777777777777B/                          READNUM
      COMMAND(1)=5HABORT                                                    READNUM
      COMMAND(2)=4HZERO                                                     READNUM
      COMMAND(3)=8HCONTINUE                                                 READNUM
      COMMAND(4)=4HREAD                                                     READNUM
      COMMAND(5)=4HLIST
      COMMAND(6)=6HREWIND
      AM1=SHIFT(MASK(3),3)
      IROW=ICOL=POINTER=II=1
      IN(73)=-1
      INUNIT=5
      END=LIST=.FALSE.
      NN=N*M
      COMMA=1R,
      IF(NAME.EQ.10HDIMENSIONI .OR. NAME.EQ.10H          A) GO TO 1
      PRINT(DSP,97)AND(NAME,77B),NAME,N,M
    1 IF(REAL) GO TO 2
      CALL SUBREL(MATRIX,MATRIX,N,M,ND)
    2 CALL HAZEL(4HBELL)
      READ(INUNIT,99)(IN(K),K=1,72)
      GO TO (3,5,7,700,9) INUNIT
    3 IF(EOF(5LTAPE1).EQ.0) GO TO 9
      GO TO 8
    5 IF(EOF(5LTAPE2).EQ.0) GO TO 9
      GO TO 8
    7 IF(EOF(5LTAPE3).EQ.0) GO TO 9
    8 INUNIT=5
      PRINT(DSP,98)
      GO TO 2
    9 KK=1
      K=0
   10 K=K+1
      GO=AND(SHIFT(A(IN(K)+2),-3*(II-1)),AM1)
      GO TO (10,200,300,400,520,600,700) GO
   97 FORMAT(/,5X,A=,"   DIMENSIONED (",I3,",",I3,") IS OR[illegible]",/)
   98 FORMAT(/30X,"EOF ENCOUNTERED ON TAPE, ENTER REMAININ[illegible]",/,
     ^30X,"DATA FROM KEYBOARD",/)
   99 FORMAT(72R1)
C**********
C           DECODING NUMERICAL ENTRIES
```

```
C**********                                                                       READNUM
  200 IF(IN(K).EQ.1R*) GO TO 247                                                  READNUM
      IF(POINTER.GT.NN .AND. END) GO TO 249                                       READNUM
      NE=0                                                                        READNUM
      NUMST=K                                                                     READNUM
      I=1                                                                         READNUM
      IF(IN(K).EQ.1R.) I=2                                                        READNUM
  210 K=K+1                                                                       READNUM
      IF(I.EQ.3) I=4                                                              READNUM
  215 GO=AND(SHIFT(A(IN(K)+2),-3*I),AM1)                                          READNUM
      GO TO (210,220,230,240,250,260,270) GO                                      READNUM
  220 IF(IN(K-1).LE.36) GO TO 225                                                 READNUM
      IF(K-2.EQ.0) GO TO 500                                                      READNUM
      IF(IN(K-2).LE.36) GO TO 225                                                 READNUM
      GO TO 500                                                                   READNUM
  225 I=3                                                                         READNUM
      K=K+1                                                                       READNUM
      NE=K                                                                        READNUM
      GO TO 215                                                                   READNUM
  230 I=2                                                                         READNUM
      GO TO 210                                                                   READNUM
  240 IF(I.EQ.3) GO TO 500                                                        READNUM
      IF(I.EQ.4) GO TO 241                                                        READNUM
      IF(IN(K-1).LE.36) GO TO 245                                                 READNUM
      IF(K-2.EQ.0) GO TO 500                                                      READNUM
      IF(IN(K-2).LE.36) GO TO 245                                                 READNUM
      GO TO 500                                                                   READNUM
  241 L=0                                                                         READNUM
      IF(IN(NE).EQ.1R+ .OR. IN(NE).EQ.1R-) L=1                                    READNUM
      IF(K-NE-3-L)245,242,500                                                     READNUM
  242 IF(IN(NE+L)-29)245,243,500                                                  READNUM
  243 IF(IN(NE+L+1)-36)245,244,500                                                READNUM
  244 IF(IN(NE+L+2).GT.1R3) GO TO 500                                             READNUM
  245 REWIND 7                                                                    READNUM
      L=K-1                                                                       READNUM
      WRITE(7,99)(IN(J),J=NUMST,L)                                    READNUM   202
      REWIND 7                                                        READNUM   203
      READ(7,*)MATRIX(IROW,ICOL)                                      READNUM   204
  247 POINTER=POINTER+1                                               READNUM   205
      IROW=(POINTER-1)/M+1                                            READNUM   206
      ICOL=POINTER-(IROW-1)*M                                         READNUM   207
      IF(POINTER.GE.NN+1) GO TO 249                                   READNUM   208
  248 KK=K                                                            READNUM   209
      IF(IN(K).EQ.1R*) KK=K+1                                         READNUM   210
      GO TO 10                                                        READNUM   211
  249 END=.TPUE.                                                      READNUM   212
      II=8                                                            READNUM   213
      GO TO 248                                                       READNUM   214
  250 GO TO 520                                                       READNUM   215
  260 GO TO 560                                                       READNUM   216
  270 GO TO 300                                                       READNUM   217
C**********                                                           READNUM   218
C             DFCODES COMMANDS                                        READNUM   219
C**********                                                           READNUM   220
  300 REW=1                                                           READNUM   221
      IF(IN(K).EQ.1R$) GO TO 330                                      READNUM   222
  305 WORDST=K                                                        READNUM   223
      I=6                                                             READNUM   224
      L=-1                                                            READNUM   225
  310 K=K+1                                                           READNUM   225
      L=L+1                                                           READNUM   227
      GO=AND(SHIFT(A(IN(K)+2),-3*I),AM1)                              READNUM   228
      IF(GO.EQ.5) GO TO 330                                           READNUM   229
      IF(GO.EQ.2) GO TO 320                                           READNUM   230
      IF(GO.EQ.3) GO TO 520                                           READNUM   231
      IF(L.GT.9) GO TO 530                                            READNUM   232
      IF(GO.EQ.6) GO TO 560                                           READNUM   233
      GO TO 310                                                       READNUM   234
  320 WORD=IN(WORDST)                                                 READNUM   235
      IF(L.EQ.0) GO TO (324,366) REW                                  READNUM   236
      DO 322 J=1,L
  322 WORD=OR(SHIFT(WORD,6),IN(WORDST+J))
      IF(REW.EQ.2) GO TO 366
  324 DO 325 J=1,5
      IF(AND(SHIFT(COMMAND(J),6*(L+1)),AM(L+1)).EQ.WORD) GO TO 327
  325 CONTINUE
      GO TO 530
  327 IF(IN(K).EQ.1R/ .AND. J.NE.5) GO TO 540
      GO TO (330,340,350,360,370) J
  330 IF(REAL) RETURN ABORT
      DO 335 J=1,N
      DO 335 JJ=1,M
  335 IF(ABS(MATRIX(J,JJ)).GT.1E14) MATRIX(J,JJ)=0.0
      CALL SUBINT(MATRIX,MATRIX,N,M,ND)
      RETURN ABORT
  340 DO 345 J=1,N
      DO 345 JJ=1,M
  345 MATRIX(J,JJ)=0.0
      END=.FALSE.
      IROW=ICOL=POINTER=1
      PRINT(DSP,399)AND(NAME,AM(1)),NAME
      KK=K
      GO TO 10
  350 END=.TRUE.
      GO TO 600
  360 K=K+2
      NUM=0
      IF(IN(K-1).EQ.28) NUM=1
      IF(IN(K-1).EQ.29) NUM=2
      IF(IN(K-1).EQ.30) NUM=3
      IF(IN(K-1).EQ.32) NUM=5
      IF(NUM.EQ.0) GO TO 540
      IF(IN(K).NE.45 .AND. IN(K).NE.46) GO TO 540
      IF(IN(K+1).EQ.1RR) GO TO 365
  364 INUNIT=NUM
```

```
      GO TO 2                                                             READNUM    272
  365 REW=2                                                               READNUM    273
      K=K+1                                                               READNUM    274
      GO TO 305                                                           READNUM    275
  366 IF(AND(SHIFT(COMMAND(6),6*(L+1)),AM(L+1)).NE.WORD)  GO TO 530       READNUM    276
      REWIND NUM                                                          READNUM    277
      GO TO 364                                                           READNUM    278
  370 IF(IN(K).NE.1R/ .AND. IN(K+1).NE.1R/) GO TO 372                     READNUM    279
      IF(IN(K).NE.1R/) K=K+1                                              READNUM    280
      LIST=.TRUE.                                                         READNUM    281
      GO TO 400                                                           READNUM    282
  372 SLASH(1)=SLASH(2)=0                                                 READNUM    283
  375 LIST=.FALSE.                                                        READNUM    284
      IF(SLASH(2).GT.M .OR. SLASH(1).GT.NN)  GO TO 550                    READNUM    285
      IF(SLASH(2).NE.0 .AND. SLASH(1).GT.N)  GO TO 550                    READNUM    286
      IF(M.EQ.1 .AND.N.EQ.1)  GO TO 380                                   READNUM    287
      IF(M.EQ.1)  GO TO 381                                               READNUM    288
      IF(N.EQ.1)  GO TO 383                                               READNUM    289
      IF(SLASH(1).EQ.0 .AND. SLASH(2).EQ.0)  GO TO 388                    READNUM    290
      IF(SLASH(1).EQ.0)  GO TO 386                                        READNUM    291
      IF(SLASH(2).EQ.0)  GO TO 387                                        READNUM    292
      PRINT(DSP,390)AND(NAME,AM(1)),NAME,SLASH(1),SLASH(2),MATRIX(SLASH(  READNUM    293
     ^1),SLASH(2))                                                        READNUM    294
  377 KK=K+1                                                              READNUM    295
      END=.FALSE.                                                         READNUM    295
      GO TO 10                                                            READNUM    297
  380 PRINT(DSP,398)AND(NAME,AM(1)),NAME,MATRIX(1,1)                      READNUM    298
      GO TO 377                                                           READNUM    299
  381 IF(SLASH(1).EQ.0)  GO TO 382                                        READNUM    300
      PRINT(DSP,397)AND(NAME,AM(1)),NAME,SLASH(1),MATRIX(SLASH(1),1)      READNUM    301
      GO TO 377                                                           READNUM    302
  382 J=1                                                                 READNUM    303
      PRINT(DSP,396)AND(NAME,AM(1)),NAME,J                                READNUM    304
      PRINT(DSP,395) (MATRIX(JJ,J),JJ=1,N)                                READNUM    305
      PRINT(DSP,*)                                                        READNUM    306
      GO TO 377                                                      READNUM   307
  383 IF(SLASH(2).EQ.0 .AND. SLASH(1).GT.1) GO TO 384                READNUM   308
      IF(SLASH(2).EQ.0) GO TO 385                                    READNUM   309
      PRINT(DSP,397)AND(NAME,AM(1)),NAME,SLASH(2),MATRIX(1,SLASH(2)) READNUM   310
      GO TO 377                                                      READNUM   311
  384 PRINT(DSP,397)AND(NAME,AM(1)),NAME,SLASH(1),MATRIX(1,SLASH(1)) READNUM   312
      GO TO 377                                                      READNUM   313
  385 J=1                                                            READNUM   314
      PRINT(DSP,394)AND(NAME,AM(1)),NAME,J                           READNUM   315
      PRINT(DSP,393)(MATRIX(J,JJ),JJ=1,M)                            READNUM   316
      PRINT(DSP,*)                                                   READNUM   317
      GO TO 377                                                      READNUM   318
  386 PRINT(DSP,396)AND(NAME,AM(1)),NAME,SLASH(2)                    READNUM   319
      PRINT(DSP,395)(MATRIX(J,SLASH(2)),J=1,N)                       READNUM   320
      PRINT(DSP,*)                                                   READNUM   321
      GO TO 377                                                      READNUM   322
  387 PRINT(DSP,394)AND(NAME,AM(1)),NAME,SLASH(1)                    READNUM   323
      PRINT(DSP,393)(MATRIX(SLASH(1),J),J=1,M)                       READNUM   324
      PRINT(DSP,*)                                                   READNUM   325
      GO TO 377                                                      READNUM   326
  388 PRINT(DSP,392)AND(NAME,AM(1)),NAME                             READNUM   327
      DO 389 J=1,N                                                   READNUM   328
  389 PRINT(DSP,391)J,(MATRIX(J,JJ),JJ=1,M)                          READNUM   329
      PRINT(DSP,*)                                                   READNUM   330
      GO TO 377                                                      READNUM   331
  390 FORMAT(5X,A=,"(",I3,",",I3,") = ",G20.14,/)                    READNUM   332
  391 FORMAT(/1X,I3,")",G15.8,3(2X,G15.8),/,4(2X,G15.8))             READNUM   333
  392 FORMAT(25X,A=)                                                 READNUM   334
  393 FORMAT(2X,4(3X,G15.8))                                         READNUM   335
  394 FORMAT(5X,A=,", ROW ",I3,/)                                    READNUM   336
  395 FORMAT(5X,G15.8)                                               READNUM   337
  396 FORMAT(5X,A=,", COLUMN ",I3,/)                                 READNUM   338
  397 FORMAT(5X,A=,"(",I3,") = ",G20.14,/)                           READNUM   339
  398 FORMAT(5X,A=," = ",G20.14,/)                                   READNUM   340
  399 FORMAT(/,30X,A=," SET TO ZERO",//)                             READNUM   341
```

```
C**********                                                     READNUM   342
C             DECODES INDECIES WITHIN SLASHES                   READNUM   343
C**********                                                     READNUM   344
  400 SLASH(1)=SLASH(2)=0                                       READNUM   345
      I=5                                                       READNUM   346
      K=K+1                                                     READNUM   347
      GO=AND(SHIFT(A(IN(K)+2),-3*I),AM1)                        READNUM   348
      GO TO (410,420,430,440,520,560,300) GO                    READNUM   349
  410 PET=1                                                     READNUM   350
      GO TO 480                                                 READNUM   351
  411 SLASH(1)=NUM                                              READNUM   352
      IF(IN(K).EQ.1R/) GO TO 440                                READNUM   353
      K=K+1                                                     READNUM   354
      IF(IN(K).EQ.1R/) GO TO 440                                READNUM   355
      IF(IN(K).EQ.45 .AND. IN(K-1).NE.46) GO TO 540             READNUM   356
      IF(IN(K).EQ.45) GO TO 415                                 READNUM   357
      IF(IN(K).EQ.1R$) GO TO 300                                READNUM   358
      IF(AND(SHIFT(A(IN(K)+2),-3*I),AM1).NE.1) GO TO 540        READNUM   359
      RET=2                                                     READNUM   360
      GO TO 480                                                 READNUM   361
  412 SLASH(2)=NUM                                              READNUM   362
      K=K-1                                                     READNUM   363
  415 K=K+1                                                     READNUM   364
      IF(IN(K).EQ.1R/) GO TO 440                                READNUM   365
      IF(IN(K).EQ.1R$) GO TO 300                                READNUM   366
      GO TO 540                                                 READNUM   367
  420 K=K+1                                                     READNUM   368
      IF(IN(K).EQ.1R/) GO TO 440                                READNUM   369
      IF(IN(K).NE.46) GO TO 435                                 READNUM   370
  430 K=K+1                                                     READNUM   371
  435 IF(IN(K).EQ.1R$) GO TO 300                                READNUM   372
      IF(IN(K).EQ.-1) GO TO 560                                 READNUM   373
      IF(AND(SHIFT(A(IN(K)+2),-3*I),AM1).NE.1) GO TO 540        READNUM   374
      RET=2                                                     READNUM   375
      GO TO 480                                                 READNUM   376
```

```
440 IF(LIST) GO TO 375                                            READNUM
    IF(SLASH(1).EQ.0 .AND. SLASH(2).EQ.0) GO TO 460               READNUM
    IF(SLASH(1).EQ.0) GO TO 540                                   READNUM
    IF(SLASH(2).EQ.0) GO TO 450                                   READNUM
    IF(SLASH(1).GT.N .OR. SLASH(2).GT.M) GO TO 550                READNUM
    IROW=SLASH(1)                                                 READNUM
    ICOL=SLASH(2)                                                 READNUM
    POINTEP=(IROW-1)*M+ICOL                                       READNUM
    END=.FALSE.                                                   READNUM
445 KK=K+1                                                        READNUM
    II=1                                                          READNUM
    GO TO 10                                                      READNUM
450 IF(SLASH(1).GT.NN) GO TO 550                                  READNUM
    POINTEP=SLASH(1)                                              READNUM
    IROW=(POINTER-1)/M+1                                          READNUM
    ICOL=POINTER-(IROW-1)*M                                       READNUM
    END=.FALSE.                                                   READNUM
    GO TO 445                                                     READNUM
460 IF(N.EQ.1 .AND. M.EQ.1) GO TO 464                             READNUM
    IF(N.EQ.1) GO TO 465                                          READNUM
    IF(M.EC.1) GO TO 467                                          READNUM
    PRINT(DSP,499)AND(NAME,AM(1)),NAME,IROW,ICOL                  READNUM
462 KK=K+1                                                        READNUM
    GO TO 10                                                      READNUM
464 PRINT(DSP,497)AND(NAME,AM(1)),NAME                            READNUM
    GO TO 462                                                     READNUM
465 PRINT(DSP,498)AND(NAME,AM(1)),NAME,ICOL                       READNUM
    GO TO 462                                                     READNUM
467 PRINT(DSP,498)AND(NAME,AM(1)),NAME,IROW                       READNUM
    GO TO 462                                                     READNUM
480 NUMST=K                                                       READNUM
485 K=K+1                                                         READNUM
    IF(AND(SHIFT(A(IN(K)+2),-3*I),AM1).EQ.1) GO TO 485            READNUM
    IF(IN(K).EQ.1R$) GO TO 300                                    READNUM
    IF(IN(K).EQ.-1) GO TO 560                                     READNUM
```

```
      IF(IN(K).NE.45 .AND. IN(K).NE.46 .AND. IN(K).NE. 1R/) GO TO 520     READNUM
      L=K-1                                                                READNUM
      REWIND 7                                                             READNUM
      WRITE(7,99)(IN(J),J=NUMST,L)                                         READNUM
      REWIND 7                                                             READNUM
      READ(7,*)TEMP                                                        READNUM
      IF(TEMP.LT.0.0) GO TO 550                                            READNUM
      NUM=IFIX(TEMP)                                                       READNUM
      GO TO (411,412) RET                                                  READNUM
  497 FORMAT(/,30X,A=," IS A NONDIMENSIONED VARIABLE",/)                   READNUM
  498 FORMAT(/,30X,"NEXT ENTRY GOES INTO",/,30X,A=,"(",I3,")"/)           READNUM
  499 FORMAT(/,30X,"NEXT ENTRY GOES INTO",/,30X,A=,"(",I3,",",I3,")"/)    READNUM
C**********                                                                READNUM
C            ERROR PRINTING SECTION                                        READNUM
C**********                                                                READNUM
  500 LL=KK-1                                                              READNUM
      L1=K-KK                                                              READNUM
      PRINT(DSP,599)(IN(J),J=1,72)                                         READNUM
      PRINT(DSP,596)LL,L1                                                  READNUM
      PRINT(DSP,590)                                                       READNUM
  505 IF(LL.EQ.0) GO TO 510                                                READNUM
      PRINT(DSP,599)(IN(J),J=1,LL),COMMA                                   READNUM
      GO TO 515                                                            READNUM
  510 PRINT(DSP,*)                                                         READNUM
  515 PRINT(DSP,598)KK                                                     READNUM
      GO TO 575                                                            READNUM
  520 LL=KK-1                                                              READNUM
      L=K-1                                                                READNUM
      PRINT(DSP,599)(IN(J),J=1,72)                                         READNUM
      PRINT(DSP,597)L                                                      READNUM
      PRINT(DSP,591)                                                       READNUM
      GO TO 505                                                            READNUM
  530 LL=KK-1                                                              READNUM
      L1=K-KK                                                              READNUM
      PRINT(DSP,599)(IN(J),J=1,72)                                         READNUM
      PRINT(DSP,596)LL,L1                                                 READNUM
      PRINT(DSP,592)                                                      READNUM
      GO TO 505                                                           READNUM
  540 LL=KK-1                                                             READNUM
      L1=K-KK                                                             READNUM
      PRINT(DSP,599)(IN(J),J=1,72)                                        READNUM
      PRINT(DSP,596)LL,L1                                                 READNUM
      PRINT(DSP,593)                                                      READNUM
      GO TO 505                                                           READNUM
  550 LL=KK-1                                                             READNUM
      L1=K-KK                                                             READNUM
      PRINT(DSP,599)(IN(J),J=1,72)                                        READNUM
      PRINT(DSP,596)LL,L1                                                 READNUM
      PRINT(DSP,594)                                                      READNUM
      GO TO 505                                                           READNUM
  560 LL=KK-1                                                             READNUM
      L1=K-KK                                                             READNUM
      PRINT(DSP,599)(IN(J),J=1,72)                                        READNUM
      PRINT(DSP,596)LL,L1                                                 READNUM
      PRINT(DSP,595)                                                      READNUM
      PRINT(DSP,599)(IN(J),J=1,LL),COMMA                                  READNUM
      PRINT(DSP,*)                                                        READNUM
  575 INUNIT=5                                                            READNUM
      IF(BATCH) RETURN ABORT                                              READNUM
      LIST=.FALSE.                                                        READNUM
      II=1                                                                READNUM
      GO TO 2                                                             READNUM
  590 FORMAT(/,30X,"ILL FORMATED NUMBER--PLEASE RETYPE FROM:",//)         READNUM
  591 FORMAT(/,30X,"ILLEGAL CHARACTER--PLEASE RETYPE FROM:",//)           READNUM
  592 FORMAT(/,30X,"COMMAND UNRECOGNIZED--PLEASE RETYPE FROM:",//)        READNUM
  593 FORMAT(/,30X,"FORMAT ERROR--PLEASE RETYPE FROM:",//)                READNUM
  594 FORMAT(/,30X,"INDICE OUT OF RANGE--PLEASE RETYPE FROM:",//)         READNUM
  595 FORMAT(/,30X,"MORE THAN 72 CHARACTERS--RETYPE ALL AFTER:",//)       READNUM
  596 FORMAT(1X,=X,=("^"))                                                READNUM
  597 FORMAT(1X,=X,"^")                                                   READNUM
  598 FORMAT("+",=X)                                                  READNUM    482
  599 FORMAT(1X,72R1)                                                 READNUM    483
C**********                                                           READNUM    484
C          TERMINATION OF READNUM OPERATION                           READNUM    485
C**********                                                           READNUM    486
  600 IF(END) GO TO 650                                               READNUM    487
      GO TO 2                                                         READNUM    488
  650 IF(REAL) RETURN                                                 READNUM    489
      DO 660 J=1,N                                                    READNUM    490
      DO 660 JJ=1,M                                                   READNUM    491
      IF(ABS(MATRIX(J,JJ)).GT.1E14) GO TO 670                         READNUM    492
  660 CONTINUE                                                        READNUM    493
      CALL SUPINT(MATRIX,MATRIX,N,M,ND)                               READNUM    494
      RETURN                                                          READNUM    495
  670 PRINT(DSP,699)AND(NAME,AM(1)),NAME,J,JJ,MATRIX(J,JJ)            READNUM    496
      IROW=J                                                          READNUM    497
      ICOL=JJ                                                         READNUM    498
      POINTER=(IROW-1)*M+ICOL                                         READNUM    499
      GO TO 575                                                       READNUM    500
  699 FORMAT(30X,A=," IS INTEGER MODE:",/,30X,"ELEMENT(",I3,",",I3,   READNUM    501
     ^") = ",G16.10,/,30X,"IS TOO LARGE--PLEASE REENTER",/)           READNUM    502
C**********                                                           READNUM    503
C          GLITCH IN PROGRAM                                          READNUM    504
C**********                                                           READNUM    505
  700 PRINT(DSP,799)                                                  READNUM    506
      RETURN ABORT                                                    READNUM    507
  799 FORMAT(72("*"),/,3X,"PROGRAM IMPROPERLY LOADED--TERMINATE AND RELO READNUM    508
     ^AD",/,72("*"))                                                  READNUM    509
      END                                                             READNUM    510
      SUBROUTINE SUBINT(DAT,IOAT,N,M,NO)                             READNUM   511
      DIMENSION DAT(ND,M),IDAT(ND,M)                                 READNUM   512
      DO 10 J=1,N                                                    READNUM   513
      DO 10 JJ=1,M                                                   READNUM   514
      IDAT(J,JJ)=IFIX(DAT(J,JJ))                                     READNUM   515
   10 CONTINUE                                                       READNUM   516
      RETURN                                                         READNUM   517
      ENTRY SUBREL                                                   READNUM   518
      DO 20 J=1,N                                                    READNUM   519
      DO 20 JJ=1,M                                                   READNUM   520
   20 DAT(J,JJ)=FLOAT(IDAT(J,JJ))                                    READNUM   521
      RETURN                                                         READNUM   522
      END                                                            READNUM   523
      SUBROUTINE READCOM(COM,NUM)                                       READCOM    2
      INTEGER COM(NUM),IN(80),GO,ANS,DSP,AM(10),DUP                     READCOM   67
      COMMON/CONTROL/DSP,BATCH                                          READCOM   68
      COMMON/LISTING/NL,LIST(1)                                         READCOM   69
      LOGICAL COMMA,BATCH                                               READCOM   70
      DATA AM/77B,7777B,777777B,77777777B,7777777777B,                  READCOM   71
     ^777777777777B,77777777777777B,7777777777777777B,                  READCOM   72
     ^777777777777777777B,77777777777777777777B/                        READCOM   73
    5 DO 10 I=1,NUM                                                     READCOM   74
   10 COM(I)=0                                                          READCOM   75
      J=0                                                               READCOM   76
      COMMA=.FALSE.                                                     READCOM   77
   20 READ 99,(IN(K),K=1,80)                                            READCOM   78
      KK=1                                                              READCOM   79
      GO=0                                                              READCOM   80
      DO 40 K=1,80                                                      READCOM   81
      IF(IN(K).EQ.1R$) GO TO 88                                         READCOM   82
      IF(IN(K).EQ.1R,) GO=1                                             READCOM   83
      IF(IN(K).EQ.1R ) GO=2                                             READCOM   84
      IF(IN(K).EQ.1R/) GO=3                                             READCOM   85
      IF(GO.EQ.0) GO TO 40                                              READCOM   86
      LL=K-KK                                                           READCOM   87
      IF(LL.GT.9) GO TO 50                                              READCOM   88
      IF(LL.EQ.0.AND.GO.EQ.3) COMMA=.FALSE.                             READCOM   89
      IF(LL.EQ.0) GO TO 35                                              READCOM   90
      IF(GO.NE.2) GO TO 25                                              READCOM   91
      IF(IN(K+1).NE.1R ) GO TO 36                                       READCOM   92
   25 J=J+1                                                             READCOM   93
      IF(J.GT.NUM) GO TO 55                                             READCOM   94
      COM(J)=0                                                          READCOM   95
      DO 30 L=1,LL                                                      READCOM   96
   30 COM(J)=OR(SHIFT(COM(J),6),IN(KK+L-1))                             READCOM   97
      IF(COM(J).NE.3R***) GO TO 31                                      READCOM   98
      J=J-1                                                             READCOM   99
      GO TO 20                                                          READCOM  100
   31 CONTINUE
      DUP=0
      DO 32 I=1,NL
      IF(COM(J).NE.AND(SHIFT(LIST(I),LL*6),AM(LL))) GO TO 32
      DUP=DUP+1
      IF(DUP.GT.1) GO TO 60
      ICOM=I
   32 CONTINUE
      IF(DUP.EQ.0) GO TO 65
      COM(J)=ICOM
      IF(COMMA) COM(J)=-ICOM
      COMMA=.FALSE.
      IF(GO.EQ.1) COMMA=.TRUE.
   35 KK=K+1
   36 GO=0
   40 CONTINUE
      IF(KK.NE.81) GO TO 70
      NUM=J
      RETURN
C*****ERROR  MORE THAN 9 CHARACTERS IN COMMAND NAME***
   50 K=KK+9
      PRINT(DSP,98)(IN(L),L=KK,K)
      GO TO 75
C*****ERROR  TOO MANY COMMAND NAMES ENTERED**
   55 K=K-1
      PRINT(DSP,97)NUM,LL,(IN(L),L=KK,K)
   56 PRINT(DSP,96)
      IF(BATCH) GO TO 88
      READ(5,99)ANS
      IF(ANS.EQ.1RY) RETURN
      IF(ANS.EQ.1R$) GO TO 88
      IF(ANS.NE.1RN) GO TO 56
      PRINT(DSP,95)
      GO TO 5
C*****ERROR  NON UNIQUE ABBREVIATION USED**
```

```
  60 J=J-1
     K=K-1
     PRINT(DSP,94)LL,(IN(L),L=KK,K),AND(LIST(ICOM),AM(1)),LIST(ICOM),
    1  AND(LIST(I),AM(1)),LIST(I)
     GO TO 75
C*****ERROR  COMMAND NAME UNRECOGNIZED**
  65 J=J-1
     K=K-1
     PRINT(DSP,93)LL,(IN(L),L=KK,K)
     GO TO 75
C*****ERROR  INCOMPLETE COMMAND DUE INPUT CHARACTER COUNT LIMITATION**
  70 PRINT(DSP,92)
  75 IF(J.EQ.0) GO TO 80
     ICOM=IABS(COM(J))
     PRINT(DSP,91)AND(LIST(ICOM),AM(1)),LIST(ICOM)
     GO TO 85
  80 PRINT(DSP,95)
  85 IF(BATCH) GO TO 88
     GO TO 20
C*****ABORT DUE TO $ OR ERROR IN BATCH MODE**
  88 NUM=COM(1)=0
     RETURN
  91 FORMAT(/5X,"PLEASE RETYPE ALL AFTER -   ",A=,/)
  92 FORMAT(/10X,"COMMAND TRUNCATED DUE TO CHARACTER COUNT.")
  93 FORMAT(/1X,=R1,"  - IS AN INVALID COMMAND.")
  94 FORMAT(/1X,=R1,"  IS A NON UNIQUE ABBREVIATION (",A=,
    1  " OR ",A=,"???).")
  95 FORMAT(/5X,"PLEASE REENTER COMMAND >>")
  96 FORMAT(20X,"TYPE    YES -- TO PROCESS CURRENT COMMANDS",/,
    1  27X,"NO -- TO REENTER ALL COMMANDS      >>")
  97 FORMAT(/10X,"MORE THAN ",I2," COMMANDS ENTERED;  ",=R1,
    1  " AND ALL AFTER",/,10X,"WILL BE IGNORED.")
  98 FORMAT(/1X,10R1,"...  IS ILLEGAL (MORE THAN 9 CHARACTERS).")
  99 FORMAT(80R1)
     END
```

```
      SUBROUTINE PRINTR(NA,NAM,VAR,N,M,ND)
      COMMON/LISTOUT/NL,LISTO(1)
      COMMON/TITLE/ITITLF(5)
      COMMON/CONTROL/DSP
      INTEGEF DSP
      NI=1RE
      GO TO 10
      ENTRY PRINTI
      NI=1RI
   10 IND=3
      IF(NA.LE.0 .OR. NA.GT.NL) GO TO 15
      IND=LISTO(NA)
      IF(IND.EQ.1) GO TO 38
   15 NAME=NAM
      DO 20 I=1,5
      IF(ITITLE(I).NE.1H ) GO TO 25
   20 CONTINUE
      GO TO 30
   25 NAME=10H         A
   30 NR=1
      NDST=6
      IF(IND.EQ.2) NDST=DSP
      IF(IND.EQ.3) NR=2
      DO 35 J=1,NR
      PRINT(NDST,50)ITITLE
      CALL LISTER(NAME,VAR,N,M,ND,NI,NDST)
   35 NDST=DSP
   38 DO 40 I=1,5
   40 ITITLE(I)=1H
      RETURN
   50 FORMAT("N",5X,5A10)
      END
      SUBROUTINE LISTER(NAME,VAR,N,M,ND,FOR,DST)                        LISTER      2
      COMMON/FORMS/NDREP,NDFW,NDNSD,NOREP,NOFW,NONSD                    LISTER      3
      COMMON/CONTROL/DSP                                                LISTER      4
      INTEGER DST,DSP                                                   LISTER      5
      DIMENSION VAR(ND,M)                                               LISTER      6
      NUM=NOREP                                                         LISTER      7
      IFW1=NOFW                                                         LISTER      8
      NSD1=NONSD                                                        LISTER      9
      IF(DST.NE.DSP) GO TO 10                                           LISTER     10
      NUM=NDREP                                                         LISTER     11
      IFW1=NDFW                                                         LISTER     12
      NSD1=NDNSD                                                        LISTER     13
   10 CONTINUE                                                          LISTER     14
      IF(M.EQ.1 .AND. N.EQ.1) GO TO 100                                 LISTER     15
      IF(M.EQ.1) GO TO 200                                              LISTER     16
      IF(N.EQ.1) GO TO 300                                              LISTER     17
      GO TO 400                                                         LISTER     18
  100 PRINT(DST,110)AND(NAME,77B),NAME,FOR,IFW1,NSD1,VAR(N,M)           LISTER     19
      GO TO 500                                                         LISTER     20
  110 FORMAT(1P,/,10X,A=," = ",V=.=,/)                                  LISTER     21
  200 PRINT(DST,210)AND(NAME,77B),NAME                                  LISTER     22
      PRINT(DST,211)((FOR,IFW1,NSD1,VAR(J,1)),J=1,N)                    LISTER     23
      GO TO 500                                                         LISTER     24
  210 FORMAT(1P,/,10X,A=,/)                                             LISTER     25
  211 FORMAT(1P,6X,V=.=)                                                LISTER     26
  300 PRINT(DST,310)AND(NAME,77B),NAME                                  LISTER     27
      MM=M                                                              LISTER     28
      IF(M.GT.NUM) MM=NUM                                               LISTER     29
      PRINT(DST,311)NUM,((FOR,IFW1,NSD1,VAR(1,J)),J=1,MM)               LISTER     30
      IF(M.GT.NUM) GO TO 406                                            LISTER     31
      GO TO 500                                                         LISTER     32
  310 FORMAT(1P,/,10X,A=,/)                                             LISTER     33
  311 FORMAT(1P,4X,=(1X,V=.=))                                          LISTER     34
  400 PRINT(DST,410)AND(NAME,77B),NAME                                  LISTER     35
      MM=M                                                              LISTER     35
      IF(M.GT.NUM) MM=NUM                                               LISTER
      DO 405 J=1,N                                                      LISTER
  405 PRINT(DST,411)J,FOR,IFW1,NSD1,VAR(J,1),NUM-1,                     LISTER
     1  ((FOR,IFW1,NSD1,VAR(J,K)),K=2,MM)                               LISTER
      IF(M.LE.NUM) GO TO 500                                            LISTER
  406 MST=NUM+1                                                         LISTER
  407 MM=M-MST+1                                                        LISTER
      IF(MM.GT.NUM) MM=NUM                                              LISTER
      MEND=MST+MM-1                                                     LISTER
      PRINT(DST,*)                                                      LISTER
      PRINT(DST,*)                                                      LISTER
      DO 408 J=1,N                                                      LISTER
  408 PRINT(DST,412)NUM,((FOR,IFW1,NSD1,VAR(J,K)),K=MST,MEND)           LISTER
      IF(MEND.EQ.M) GO TO 500                                           LISTER
      MST=MEND+1                                                        LISTER
      GO TO 407                                                         LISTER
  410 FORMAT(1P,/,10X,A=,/)                                             LISTER
  411 FORMAT(1P,1X,I3,"}  ",V=.=,=(1X,V=.=))                            LISTER
  412 FORMAT(1P,3X,=(1X,V=.=))                                          LISTER
  500 PRINT(DST,510)                                                    LISTER
  510 FORMAT(//)                                                        LISTER
      RETURN                                                            LISTER
      END                                                               LISTER
      SUBROUTINE OPEN(NS),RETURNS(ABORT)
      COMMON/LISTING/NL,LIST(75,7)
      COMMON/LISTOUT/NO,LISTO(1)
      COMMON/RVAR/NTOT,TOT(1)
      COMMON/IVAR/NITOT,ITOT(1)
      INTEGER DIM(2)
      IF(LIST(NS,1).EQ.10HDIMENSIONI) PRINT(8,10)
   10 FORMAT(/,5X,"PLANT DIMENSIONS (N,M,NCC,NDD) IS OPEN",/)
      IF(LIST(NS,6).NE.1 .OR. LIST(NS,7).NE.1) GO TO 25
      ENTRY FEOPEN
      IF(LIST(NS,2).EQ.3HREL) GO TO 50
      GO TO 75
   25 PRINT(8,199)AND(LIST(NS,1),77B),LIST(NS,1),LIST(NS,6),LIST(NS,7)
      DIM(1)=LIST(NS,4)
      DIM(2)=LIST(NS,5)
      CALL READNUM(10HDIMENSIONI,DIM,1,2,1,.FALSE.),RETURNS(40)
      LIST(NS,4)=DIM(1)
      LIST(NS,5)=DIM(2)
      IF(LIST(NS,4).GT.LIST(NS,6) .OR. LIST(NS,5).GT.LIST(NS,7)
     1  .OR. LIST(NS,4).LT.1 .OR. LIST(NS,5).LT.1) GO TO 25
      IF(LIST(NS,2).EQ.3HREL) GO TO 50
      GO TO 75
   40 RETURN ABORT
   50 CALL READNUM(LIST(NS,1),TOT(LIST(NS,3)),LIST(NS,4),LIST(NS,5),
     1  LIST(NS,6),.TRUE.),RETURNS(40)
      RETURN
   75 CALL READNUM(LIST(NS,1),ITOT(LIST(NS,3)),LIST(NS,4),LIST(NS,5),
     1  LIST(NS,6),.FALSE.),RETURNS(40)
      RETURN
  199 FORMAT(/,5X,"ENTER DESIRED DIMENSIONS OF ",A=,", MAX ALLOWED IS ("
     1,I3,",",I3,").",/)
      END
      SUBROUTINE FORMOUT
      COMMON/CONTROL/DSP,BATCH
      LOGICAL BATCH,STP
      INTEGER A(61,15)
      REWIND 6
      STP=.FALSE.
      ICOUNT=0
      K=1
      READ(6,10)(A(K,I),I=1,15)
      K=K+1
    5 READ(6,10)(A(K,I),I=1,15)
   10 FORMAT(1R1,13A10,1R1)
      IF(EOF(6).NE.0.0) GO TO 50
      IF(A(K,1).EQ.1RN .OR. A(K,1).EQ.1R1) GO TO 20
      IF(K.EQ.61) GO TO 20
      K=K+1
      GO TO 5
   20 K=K-1
      IF(ICOUNT+K.GT.59) A(1,1)=1R1
      IF(A(1,1).EQ.1R1)ICOUNT=0
      DO 30 I=1,K
   30 PRINT(99,10)(A(I,J),J=1,15)
      DO 40 I=1,15
   40 A(1,I)=A(K+1,I)
      ICOUNT=ICOUNT+K
      K=2
      IF(STP) GO TO 55
      GO TO 5
   50 STP=.TRUE.
      GO TO 20
   55 CALL CONNEC(6LOUTPUT)
      IF(.NOT.BATCH) STOP
      REWIND 99
      REWIND 6
  100 READ(99,10)(A(1,I),I=1,15)
```

```
      PRINT(6,10)(A(1,I),I=1,15)                                        FORMOUT   37
      IF(EOF(99).NE.0.0) STOP                                           FORMOUT   38
      GO TO 100                                                         FORMOUT   39
      END                                                               FORMOUT   40
```

```
         IDENT  RETURN                                                  RETURN     2
         ENTRY  RETURN                                                  RETURN     3
         EXT    SYS=                                                    RETURN     4
RETURN   BSSZ   1                                                       RETURN     5
         SB1    1                                                       RETURN     6
         MX4    42                                                      RETURN     7
         SA2    X1                                                      RETURN     8
         SA3    2                                                       RETURN     9
LOOP2    ZR     X3,RETURN                                               RETURN    10
         IX5    X3*X4                                                   RETURN    11
         BX5    X2-X5                                                   RETURN    12
         ZR     X5,GOTIT                                                RETURN    13
         SA3    A3+B1                                                   RETURN    14
         EQ     LOOP2                                                   RETURN    15
GOTIT    BX3    -X4*X3                                                  RETURN    16
         SA3    X3+2                                                    RETURN    17
         SA4    UNLOAD                                                  RETURN    18
         BX6    X3+X4                                                   RETURN    19
         SA6    A3                                                      RETURN    20
         EQ     RETURN                                                  RETURN    21
UNLOAD   VFD    28/0,2/3,30/0                                           RETURN    22
         END                                                            RETURN    23
```

```
      SUBROUTINE ADDMAT(A,NAD,N,M,B,NBD,C,NCD)                          MATRIX    2
      DIMENSION A(NAD,M),B(NBD,M),C(NCD,M)                              MATRIX    3
      LOGICAL SUB                                                       MATRIX    4
      SUB=.FALSE.                                                       MATRIX    5
    5 DO 10 J=1,M                                                       MATRIX    6
      DO 10 I=1,N                                                       MATRIX    7
      A(I,J)=B(I,J)+C(I,J)                                              MATRIX    8
      IF(SUB) A(I,J)=B(I,J)-C(I,J)                                      MATRIX    9
   10 CONTINUE                                                          MATRIX   10
      RETURN                                                            MATRIX   11
      ENTRY SUBMAT                                                      MATRIX   12
      SUB=.TRUE.                                                        MATRIX   13
      GO TO 5                                                           MATRIX   14
      END                                                               MATRIX   15
      SUBROUTINE TRANPOS(A,NAD,N,M,B,NBD)                              MATRIX    16
      DIMENSION A(NAD,M),B(NBD,N)                                      MATRIX    17
      LOGICAL KEY                                                      MATRIX    18
      KEY=.FALSE.                                                      MATRIX    19
    5 DO 10 J=1,M                                                      MATRIX    20
      DO 10 I=1,N                                                      MATRIX    21
      A(I,J)=B(J,I)                                                    MATRIX    22
      IF(KEY) A(I,J)=B(I,J)                                            MATRIX    23
   10 CONTINUE                                                         MATRIX    24
      RETURN                                                           MATRIX    25
      ENTRY COPYAB                                                     MATRIX    26
      KEY=.TRUE.                                                       MATRIX    27
      GO TO 5                                                          MATRIX    28
      END                                                              MATRIX    29
      SUBROUTINE PRESET(A,NAD,N,M,PRE)                            MATRIX    30
      DIMENSION A(NAD,M)                                          MATRIX    31
      LOGICAL IDT                                                 MATRIX    32
      IDT=.FALSE.                                                 MATRIX    33
    5 DO 10 J=1,M                                                 MATRIX    34
      DO 10 I=1,N                                                 MATRIX    35
      A(I,J)=PRE                                                  MATRIX    36
      IF(IDT) A(J,J)=1.0                                          MATRIX    37
   10 CONTINUE                                                    MATRIX    38
      RETURN                                                      MATRIX    39
      ENTRY IDENT                                                 MATRIX    40
      IDT=.TRUE.                                                  MATRIX    41
      IF(M.EQ.0) M=N                                              MATRIX    42
      PRE=0.0                                                     MATRIX    43
      GO TO 5                                                     MATRIX    44
      END                                                         MATRIX    45
      SUBROUTINE MATRIX1(A,NAD,N,M,B,NBD,N1,C,NCD,N2,D,NDD)             MATRIX    46
      DIMENSION A(NAD,M),B(NBD,N1),C(NCD,N2),D(NDD,M)                   MATRIX    47
      LOGICAL KEY                                                       MATRIX    48
      ENTRY MAT3MPY                                                     MATRIX    49
      KEY=.FALSE.                                                       MATRIX    50
      DO 5 J=1,M                                                        MATRIX    51
      DO 5 I=1,N                                                        MATRIX    52
    5 A(I,J)=0.0                                                        MATRIX    53
    6 DO 20 I=1,N                                                       MATRIX    54
      DO 20 K=1,N2                                                      MATRIX    55
      TEMP=0.0                                                          MATRIX    56
      DO 10 J=1,N1                                                      MATRIX    57
   10 TEMP=TEMP+B(I,J)*C(J,K)                                           MATRIX    58
      IF(KEY) A(I,K)=TEMP                                               MATRIX    59
      IF(KEY) GO TO 19                                                  MATRIX    60
      DO 15 L=1,M                                                       MATRIX    61
   15 A(I,L)=A(I,L)+TEMP*D(K,L)                                         MATRIX    62
   19 CONTINUE                                                          MATRIX    63
   20 CONTINUE                                                          MATRIX    64
      RETURN                                                            MATRIX    65
      ENTRY MAT2MPY                                                     MATRIX    66
      KEY=.TRUE.                                                        MATRIX    67
      IF(N2.EQ.0) N2=M                                                  MATRIX    68
      GO TO 6                                                           MATRIX    69
      END                                                               MATRIX    70
      SUBROUTINE INVERT(A,NAD,B,NBD,N,IFAIL)
      DIMENSION A(NAD,N),B(NBD,N)
      COMMON/CONTROL/DSP
      LOGICAL IFAIL
      INTEGER DSP
      CALL IDENT(A,NAD,N)
   35 K=0
      TEMP=0.0
      DO 40 I=J,N
      IF(TEMP.GT.ABS(B(I,J))) GO TO 40
      TEMP=ABS(B(I,J))
      K=I
   40 CONTINUE
      IF(TEMP.EQ.0.0) GO TO 90
      IF(K.EQ.J) GO TO 50
      DO 45 I=1,N
      TEMP=B(K,I)
      B(K,I)=B(J,I)
      B(J,I)=TEMP
      TEMP=A(K,I)
      A(K,I)=A(J,I)
   45 A(J,I)=TEMP
   50 IF(B(J,J).EQ.1.0) GO TO 60
      TEMP=B(J,J)
      DO 55 I=1,N
      B(J,I)=B(J,I)/TEMP
   55 A(J,I)=A(J,I)/TEMP
   60 DO 70 I=1,N
      IF(I.EQ.J) GO TO 70
      IF(B(I,J).EQ.0.0) GO TO 70
      TEMP=B(I,J)
      DO 65 K=1,N
      B(I,K)=B(I,K)-TEMP*B(J,K)
   65 A(I,K)=A(I,K)-TEMP*A(J,K)
   70 CONTINUE
```

```
      J=J+1                                                             MATRIX    106
      IF(J.LE.N) GO TO 35                                               MATRIX    107
      IFAIL=.FALSE.                                                     MATRIX    108
      RETURN                                                            MATRIX    109
   90 IFAIL=.TRUE.                                                      MATRIX    110
      PRINT(NSP,*)"  SUBROUTINE INVERT -- MATRIX IS SINGULAR"           MATRIX    111
      RETURN                                                            MATRIX    112
      END                                                               MATRIX    113
```

Appendix B

INTERAC User's Guide

INTERAC is an interactive software package for synthesizing a discrete state-variable feedback gain matrix to control a multi-input, multi-output continuous plant described by Eq (1).

$$\begin{aligned} \dot{\underline{x}}(t) &= \underline{A}\,\underline{x}(t) + \underline{B}\,\underline{u}(t) + \underline{R}\,\underline{d}(t) \\ \underline{y}(t) &= \underline{C}\,\underline{x}(t) \end{aligned} \tag{1}$$

where $\underline{A}$ is an N by N state matrix,
$\underline{B}$ is an N by M control matrix,
$\underline{R}$ is an N by NDD disturbance matrix, and
$\underline{C}$ is an (NCC or NDD) by N output matrix.

and y(t) defines the NDD outputs that are to reject disturbances or the NCC outputs that are to track input commands.

Three types of problems can be examined: (1) the regulator problem, (2) the disturbance rejector problem, and (3) the tracking problem. The computer input of the problem type and the plant description matrices can be accomplished directly by the user or in response to requests from the computer. 23 alphanumeric options are used in response to the prompt "COMMAND >>" to control data input, data output, and program sequence control.

Commands

To execute the program for the first time, type FORTRAC. All further input data will be requested by the computer. To terminate the program type STOP in response to the prompt "COMMAND >>".

For the do-it-yourself user the following commands are available for complete input,output, and program sequence control.

(1) ENTER [, parameters] This command allows the user to specify the dimensions of a matrix and then to enter the values for the matrix elements. The parameters are variable names and their use is optional. If the variable names are not included they will be requested by the computer. The legal variable names are described under options 16 through 21.

(2) STOP This command terminates the program and prints a termination message informing the user of the contents of the files ANSWER and DATA that are left by the program.

(3) END This command has two functions. First in response to the prompt "COMMAND >> ", the program is terminated, but the termination message is not printed. END is also used to terminate sub-options such as when the command ENTER is given without a parameter list.

(4) PRINT [, parameters] This command causes the current values of the variables identified in the parameter list to be printed at the terminal and to be written to the file ANSWER. The parameters are optional, and if not used the variable names will be requested by the computer.

(5) DFORMAT This command allows the user to specify the number of significant digits to be used for numbers printed at the terminal.

(6) CHANGE [, parameters] This command allows the user to change the values stored in a variable. The dimensions of matrices cannot be changed with this command. The parameters are variable names and their use is optional. If the parameters are not included the

names will be requested by the computer.

(7) DISPLAY [, parameters]   This is a dual function command. With the parameters (variable names), the variables to be printed at the terminal are specified. Without the parameters, the variables already designated to be printed at the terminal are identified and the names are printed. The values of the variables on the display list are printed at the terminal whenever the variable is first encountered or when its value is changed in the control design algorithms.

(8) OUTPUT [, parameters]   This command is used the same as DISPLAY, except OUTPUT refers to the variables to be written to the file ANSWER.

(9) DELETE [, parameters]   This command deletes the named variables (parameters) from both the display and output lists. If no parameters are given, then all variables are removed from both lists.

(10) OPTIONS   This command causes a list of the valid options to be printed at the terminal.

(11) VARIABLES   This command causes a list of the legal variable names to be printed at the terminal.

(12) SAVE   This command causes the current contents of all variables, all matrix dimensions, and the display/output list to be saved on the file DATA. The date and time of the save is printed, and they can be used to identify the data set when it is reloaded with the command RESTART.

(13) RESTART [, REWIND]   This command causes the values saved on the file DATA to be reloaded into memory. The optional parameter REWIND is used to rewind the file before reading from it.

The file DATA is a sequential file, and thus RESTART may have to be used several times to get the desired data set if SAVE was used more than once. The date and time that the data set was saved is printed to identify the set that is reloaded.

(14) REWIND This command can be used by itself to rewind the file DATA or as a parameter for RESTART as shown above.

(15) RUN ,parameter This command causes the control design algorithms to be executed starting with the subroutine named in the parameter and terminating after calculating the closed-loop matrix. The valid parameters are FORTRAC, SAMPLE , AUGMAT, TRANSFORM, CONLAW , and CLOSELOOP. All necessary data is assumed to have been entered and all matrices are assumed properly dimensioned.

(16) FORTRAC This command by itself causes all necessary data to be requested and then all control design algorithms to be executed. When this command is used as a parameter for RUN, all control design algorithms are executed with the data currently in memory.

(17) SAMPLE This command causes the sampled-data system description to be computed from the continuous system description. The variables must be entered prior to issuing this command: AMATRIX - the state matrix, BMATRIX - the control matrix, DIMENSION - the array containing the plant dimensions (N, M, NCC, NDD), RMATRIX - the disturbance matrix only if NDD is non-zero, and TSAMPLE - the desired sampling time interval. Values for the following variables are calculated: CEIGEN - the continuous eigenvalues; MODAL - the modal matrix; INVMODAL - the inverse modal matrix; and the discrete system matrices FMATRIX - the state matrix, GMATRIX - the control

matrix, DRMATRIX - the disturbance matrix if NDD is non-zero, and DEIGN - the z - plane eigenvalues.

(18) AUGMAT This command causes the sampled-data system description to be augmented with discrete integrators for disturbance rejection or tracking problems. These variables must be available in memory prior to issuing this command: FMATRIX, GMATRIX, DRMATRIX - if NDD is non-zero, CMATRIX - the output matrix if NDD or NCC is non-zero, DIMENSION, and INTEGRATE - an array containing the number of integrators desired for each NDD or NCC output. One integrator is used to track or reject a step input, two integrators for a ramp, etc. Values for the following variables are calculated: AFMATRIX- the augmented state matrix, AGMATRIX - the augmented control matrix, AHMATRIX - the augmented input command matrix, ARMATRIX - the augmented disturbance matrix, and ACMATRIX - the augmented output matrix.

(19) TRANSFORM This command causes the AF and AG matrices from the augmented system description to be transformed into the Brunovsky controllable canonical form. AFMATRIX, AGMATRIX, and DIMENSION must be available in memory prior to issuing this command. Values for the following variables are calculated: UREFORM - the upper row echelon form of the matrix [AG AF], APLFORM - the Aplevich form of the matrix [AG AF], BRNFORM - the Brunovsky form of the matrix [AG AF], BFMATRIX and BGMATRIX - the Brunovsky controllable canonical form of AFMATRIX and AGMATRIX respectively, TINVERSE - the inverse transformation matrix, and CINDICES - the controllability indices.

(20) CONLAW This command causes the state-variable feedback

gain matrix K to be calculated for the desired closed-loop eigenvalues. BFMATRIX, BGMATRIX, TINVERSE, DIMENSION, and DESEIGEN - the desired eigenvalues, must be available in memory prior to issuing this command. Values for the following variables are calculated: COFMATRIX - the coefficient matrix which will be the zero matrix unless non-zero desired eigenvalues are entered, and KMATRIX - the feedback gain matrix.

(21) CLOSELOOP This command is used to calculate the closed-loop state matrix and the closed-loop eigenvalues. The variables required in memory prior to issuing this command are: AFMATRIX, AGMATRIX, DIMENSION, and KMATRIX. CLMATRIX - the closed-loop matrix and CLEIGEN - the closed-loop eigenvalues are calculated.

(22) OFORMAT This command allows the user to specify the number of significant digits to be used for numbers written to the file ANSWER.

(23) SETDIMEN This command properly dimensions all matrices after the plant dimensions, in the array DIMENSION, have been entered.

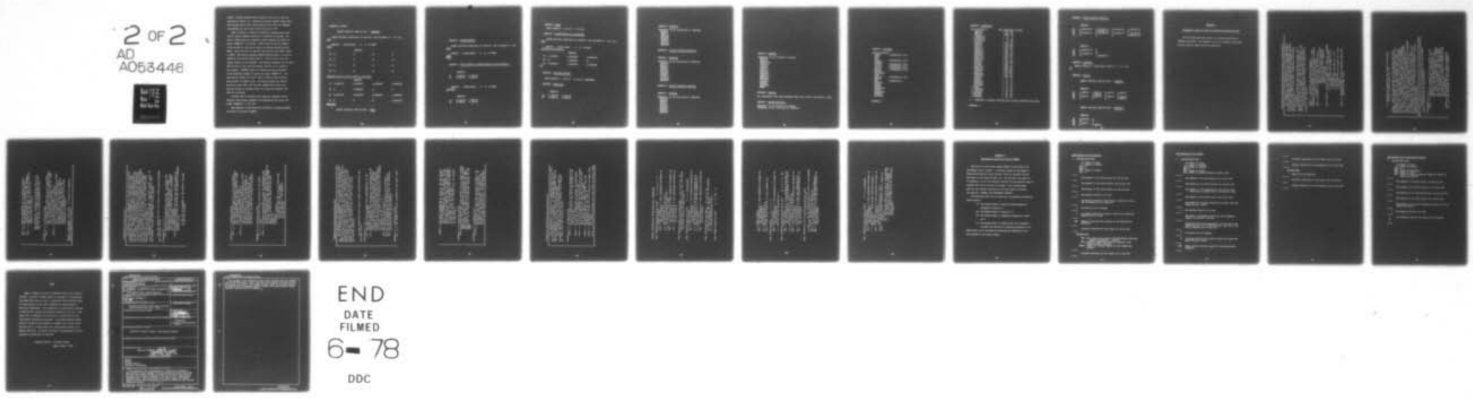
(24) "variable name" When the variable name is typed for a command, the user is allowed to set the dimensions of the variable and to enter or change the values of the elements of the variable.

(25) DIMENSION This command opens the array DIMENSION and allows the user to enter N - the number of states, M - the number of controls, and NCC - the number of commands or NDD - the number of disturbances. The elements of DIMENSION are individually addressable by the variable names STATES, CONTROLS, COMMANDS, and DISTURBS.

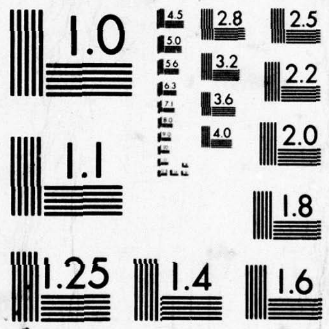
The above commands are used in response to the prompt "COMMAND >>". If the command has parameters, the parameters must be separated by

AD-A053 446   AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO   SCH--ETC   F/G 12/1
INTERAC - AN INTERACTIVE SOFTWARE PACKAGE FOR DIRECT DIGITAL CO--ETC(U)
DEC 77   J A COLGATE
UNCLASSIFIED   AFIT/GGC/EE/77D-1   NL

2 OF 2
AD
A053446



END
DATE
FILMED
6-78
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

commas. Multiple commands may be entered on one line if they are separated by slashes (/). Execution of entered commands begins after the carriage return, but to delay execution and enter more commands or parameters,the last entry on the line must be "***".

When a variable is opened for entering or changing data, there are six special commands available to facilitate the process. The first is ABORT which will terminate current commands and return the prompt "COMMAND>>" to the user. ZERO is used to set all elements of the variable to zero and is useful for entering sparse matrices. READ n allows the user to input the data from the file TAPE1, TAPE2, or TAPE3. The optional parameter REWIND following READ n, is used to rewind the file before reading from it. LIST is used to list the current contents of the variable. The optional parameter "/n,m/" after LIST is used to print only one element, one row, or one column of the variable. CONTINUE closes the variable and starts execution of any remaining commands or issues the prompt "COMMAND>>". The last special command is "/n,m/", which is used to reset the data entry pointer to element (n,m). All matrix elements are entered by rows in free format, and this last command can be used to go back and correct an erroneous entry or to skip over elements that need not be entered.

A dollar sign ($) entered at any time will terminate current activity, cause pending commands to be ignored, and will return the prompt "COMMAND>>" to the user.

The remainder of this guide will be devoted to showing examples of the use of the above commands.

```
  COMMAND >>ENTER

             ENTER VARIABLE NAME OR END  >>AMATRIX

       ENTER DESIRED DIMENSIONS OF AMATRIX, MAX ALLOWED IS ( 20, 20).
  4,4

       AMATRIX   DIMENSIONED (  4,  4) IS OPEN
  LIST
                                   AMATRIX

    1]  0.                0.                0.                0.

    2]  0.                0.                0.                0.

    3]  0.                0.                0.                0.

    4]  0.                0.                0.                0.

  1,2,3,4,/2,2/,1,/3,1/,4,3,2,1,/4,4/,5,L
                                   AMATRIX

    1]  1.0000000         2.0000000         3.0000000         4.0000000

    2]  0.                1.0000000         0.                0.

    3]  4.0000000         3.0000000         2.0000000         1.0000000

    4]  0.                0.                0.                5.0000000
  CONTINUE

             ENTER VARIABLE NAME OR END  >>END
  COMMAND >>ENTER,AMATRIX

     ENTER DESIRED DIMENSIONS OF AMATRIX, MAX ALLOWED IS < 20,
 2,2

     AMATRIX   DIMENSIONED <  2,   2> IS OPEN
 1,2
 2,3


  COMMAND >>PRINT,AMATRIX/CHANGE,AMATRIX/PRINT,AMATRIX


              AMATRIX

   1]     1.00E+00   2.00E+00
   2]     2.00E+00   3.00E+00


     AMATRIX   DIMENSIONED <  2,   2> IS OPEN
 6,7,8,9


              AMATRIX

   1]     6.00E+00   7.00E+00
   2]     8.00E+00   9.00E+00
```

```
COMMAND >>SAVE

  DATA SAVED AT  12/13/77  17.15.37.

COMMAND >>ENTER,AMATRIX/RUN,FORTRAC

  ENTER DESIRED DIMENSIONS OF AMATRIX, MAX ALLOWED IS ( 20, 20).
3,3

  AMATRIX   DIMENSIONED (  3,   3) IS OPEN
1,2,3,0,1,/3,2/,2,L

                         AMATRIX

  1]  1.0000000         2.0000000         3.0000000

  2]  0.                1.0000000         0.

  3]  4.0000000         2.0000000         2.0000000
$

COMMAND >>RESTART,REWIND

  DATA SAVED AT  12/13/77  17.15.37. RELOADED.

COMMAND >>PRINT,AM


        AMATRIX

  1]    6.00E+00   7.00E+00
  2]    8.00E+00   9.00E+00
```

```
COMMAND >>DISPLAY

VARIABLES TO BE DISPLAYED AT TERMINAL
  DEIGEN
  FMATRIX
  GMATRIX
  CLEIGEN
  KMATRIX
  TSAMPLE
  CINDICES
  BRNFORM

COMMAND >>DISPLAY,AMATRIX,BMATRIX


COMMAND >>DISPLAY

VARIABLES TO BE DISPLAYED AT TERMINAL
  AMATRIX
  BMATRIX
  DEIGEN
  FMATRIX
  GMATRIX
  CLEIGEN
  KMATRIX
  TSAMPLE
  CINDICES
  BRNFORM

COMMAND >>DELETE,TSAMPLE,CINDICES


COMMAND >>DISPLAY

VARIABLES TO BE DISPLAYED AT TERMINAL
  AMATRIX
  BMATRIX
  DEIGEN
  FMATRIX
  GMATRIX
  CLEIGEN
  KMATRIX
  BRNFORM
```

```
COMMAND >>OUTPUT

VARIABLES TO BE PRINTED TO ANSWER
  AMATRIX
  BMATRIX
  DIMENSION
  RMATRIX
  CMATRIX
  FMATRIX
  GMATRIX
  AFMATRIX
  AGMATRIX
  AEMATRIX
  ARMATRIX
  ACMATRIX
  CLEIGEN
  KMATRIX
  DESEIGEN
  MODAL

COMMAND >>DELETE

ALL VARIABLES HAVE BEEN REMOVED FROM THE DISPLAY AND OUTPUT LISTS.

COMMAND >>OUTPUT/DISPLAY

VARIABLES TO BE PRINTED TO ANSWER
VARIABLES TO BE DISPLAYED AT TERMINAL
```

```
COMMAND >>OPTIONS

VALID COMMANDS
  ENTER        [,PARAMETER LIST]
  STOP
  END
  PRINT        [,PARAMETER LIST]
  DFORMAT
  CHANGE       [,PARAMETER LIST]
  DISPLAY      [,PARAMETER LIST]
  OUTPUT       [,PARAMETER LIST]
  DELETE       [,PARAMETER LIST]
  OPTIONS
  VARIABLES
  SAVE
  RESTART      [,PARAMETER LIST]
  REWIND
  RUN          ,PARAMETER(S)
  FORTRAC
  SAMPLE
  AUGMAT
  TRANSFORM
  CONLAW
  CLOSELOOP
  DFORMAT
  SETDIMEN

COMMAND >>
COMMAND >>VARIABLES

VALID VARIABLES           MAX DIMENSION ALLOWED
     AMATRIX                   ( 20, 20)
     BMATRIX                   ( 20, 10)
     DEIGEN                    ( 20,  2)
     DIMENSION                 (  1,  1)
   ◆ COFMATRIX                 (  0,  0)
     INTEGRATE                 (  1, 10)
     RMATRIX                   ( 20,  5)
     CMATRIX                   ( 10, 20)
     FMATRIX                   ( 20, 20)
     GMATRIX                   ( 20, 10)
     DEMATRIX                  ( 20,  5)
     DRMATRIX                  ( 20,  5)
     AFMATRIX                  ( 20, 20)
     AGMATRIX                  ( 20, 10)
     AEMATRIX                  ( 20,  5)
     ARMATRIX                  ( 20,  5)
     ACMATRIX                  ( 10, 20)
     CEIGEN                    ( 20,  2)
     CLEIGEN                   ( 20,  2)
     BFMATRIX                  ( 20, 20)
     BGMATRIX                  ( 20, 10)
     TINVERSE                  ( 20, 20)
     KMATRIX                   ( 10, 20)
     DESEIGEN                  ( 20,  2)
     CLMATRIX                  ( 20, 20)
     TSAMPLE                   (  1,  1)
     STATES                    (  1,  1)
     CONTROLS                  (  1,  1)
     COMMANDS                  (  1,  1)
     DISTURBS                  (  1,  1)
   ◆ MODAL                     (  0,  0)
   ◆ INVMODAL                  (  0,  0)
   ◆ CINDICES                  (  0,  0)
   ◆ UREFORM                   (  0,  0)
   ◆ APLFORM                   (  0,  0)
   ◆ BRNFORM                   (  0,  0)

◆ -- TEMPORARY VARIABLE PRINTED ONLY DURING PROGRAM EXECUTION.

COMMAND >>
COMMAND >>PRINT,AMATRIX,BMATRIX


          AMATRIX

1]     1.00000E+00   2.00000E+00   3.00000E+00   4.00000E+00
2]     0.            1.00000E+00   0.            0.
3]     4.00000E+00   3.00000E+00   2.00000E+00   1.00000E+00
4]     0.            0.            0.            5.00000E+00



          BMATRIX

1]     3.00000E+00   0.
2]     0.            0.
3]     1.00000E+00   0.
4]     0.            1.00000E+00


COMMAND >>DFORMAT

   ENTER NUMBER OF SIGNIFICANT DIGITS   0 < N < 15.
3


COMMAND >>PRINT

          ENTER VARIABLE NAME OR END   >>AMATRIX


          AMATRIX

1]     1.00E+00   2.00E+00   3.00E+00   4.00E+00
2]     0.         1.00E+00   0.         0.
3]     4.00E+00   3.00E+00   2.00E+00   1.00E+00
4]     0.         0.         0.         5.00E+00


          ENTER VARIABLE NAME OR END   >>BMATRIX


          BMATRIX

1]     3.00E+00   0.
2]     0.         0.
3]     1.00E+00   0.
4]     0.         1.00E+00
```

Appendix C

Programmer's Guide for Using the Specialized INTERAC Software

The following pages give syntax for calling the specialized INTERAC subroutines. All elements in the call statement are defined and any required common blocks are identified.

```
      SUBROUTINE READNUM(NAME,MATRIX,N,M,ND,REAL),RETURNS(ABORT)
C***********************************************************************
C          THE PURPOSE OF THIS SUBROUTINE IS TO ALLOW (NON,SINGLE,
C     OR DOUBLE DIMENSIONED AND REAL OR INTEGER) VARIABLES TO BE
C     FILLED WITH DATA,EXAMINED, AND/OR SELECTIVE ELEMENTS CHANGED.
C     ALL ENTRIES ARE CHECKED FOR VALID FORMAT AND REENTRY IS
C     REQUESTED IF VALIDITY IS NOT MET.  NUMBERS MAY BE ENTERED
C     IN FREE FORMAT.
C          DIMENSIONED VARIABLE DATA ARE ENTERED BY ROWS, IN 72
C     COLUMN CARD IMAGES ON AS MANY LINES AS ARE REQUIRED OR
C     DESIRED TO FILL ALL ELEMENTS.  THE BELL WILL RING EACH TIME
C     THE ROUTINE IS WAITING FOR A 72 COLUMN CARD IMAGE INPUT.
C          THE FOLLOWING ALPHANUMERIC COMMANDS ARE AVAILABLE TO
C     ASSIST IN THE PROCESS.
C
C          ABORT.....TERMINATE THE ROUTINE ABNORMALLY
C          ZERO......SET ALL ELEMENTS OF THE VARIABLE TO ZERO
C          CONTINUE..ALL DATA DESIRED HAS BEEN ENTERED, EXECUTE
C                    A NORMAL RETURN FROM THE ROUTINE
C          READ N (REWIND).....READ THE DATA FOR INPUT FROM
C                              TAPE(N), WHERE N = 1, 2, 3, OR 5.
C                              REWIND IS AN OPTIONAL PARAMETER
C                              CAUSING THE TAPE TO BE REWOUND
C                              PRIOR TO READING
C          LIST (/N,M/)...'LIST' BY ITSELF WILL PRINT ALL ELEMENTS
C                         OF THE VARIABLE.  N & M CAN BE USED
C                         1) TO PRINT ONLY ONE ELEMENT (N & M
C                           SPECIFIED),
C                         2) TO PRINT ONLY ONE ROW (M = 0 OR
C                           MISSING), OR
C                         3) TO PRINT ONLY ONE COLUMN (N = 0 OR
C                           MISSING)
C          /N,M/.....SPECIFIES THAT THE NEXT NUMERIC ENTRY WILL
C                    GO INTO ELEMENT (N,M)
```

```
C         //...PRINTS THE INDICES OF THE ELEMENT INTO WHICH
C              THE NEXT NUMERIC ENTRY WILL GO
C
C    NOTES:  AFTER THE LAST ELEMENT OF THE VARIABLE HAS BEEN
C    ENTERED, THE REMAINDER OF THE 72 CHARACTER INPUT LINE WILL
C    BE SCANNED FOR ANY OF THE ABOVE COMMANDS EXCEPT READ.
C    ONLY ZERO OR /N,M/ WILL PREVENT A RETURN FROM THE ROUTINE
C    IN THIS CASE.  ABBREVIATIONS OF ANY OF THE ABOVE COMMANDS
C    ARE VALID.  AN * CAUSES THE CURRENT ELEMENT TO BE UNCHANGED
C    AND THE NEXT NUMERIC INPUT TO GO INTO THE SUCCEEDING ELEMENT.
C    A $ AT ANY TIME WILL CAUSE AN ABORT TERMINATION OF THE
C    ROUTINE.  AN ELEMENT OF A ROW OR COLUMN VECTOR CAN BE REFERENCED
C    WITH /N,M/ ( WITH THE APPROPRIATE INDICE 1) OR BY /N/.
C    ALL PRINTOUTS WILL SHOW THE ENTERED DATA IN REAL FORMAT, BUT
C    IF THE VARIABLE IS INTEGER MODE THE STORED NUMBERS WILL BE
C    CHANGED TO INTEGER FORMAT (BY TRUNCATION IF NECESSARY) BEFORE
C    THIS SUBROUTINE RETURNS.
C**********************************************************************
C**********************************************************************
C         A DEFINITION OF THE CALL PARAMENTERS AND THE REQUIRED
C    LABELED COMMON ELEMENTS IS GIVEN BELOW.
C
C    PARAMETERS----------
C         NAME......HOLLERITH CONSTANT WITH THE 'TITLE' OF THE
C                   VARIABLE IN THE FIRST 9 CHARACTERS, LEFT
C                   JUSTIFIED, AND THE NUMBER OF LETTERS IN
C                   THE 'TITLE' IN THE LAST CHARACTER.  EXAMPLE,
C                   THE TITLE OF THE VARIABLE FOR THE 'A MATRIX'
C                   MIGHT BE 10HA MATRIX H.  (THE DISPLAY CODE
C                   FOR H IS 10 WHICH IS THE OCTAL EQUIVALENT OF
C                   THE INTEGER 8 - THE NUMBER OF LETTERS IN THE
C                   TITLE 'A MATRIX')
C         MATRIX....THE VARIABLE FOR WHICH INPUT IS DESIRED.
C         N.........ROW DIMENSION OF THE VARIABLE DESIRED.
```

```
C          M.........COLUMN DIMENSION OF THE VARIABLE DESIRED.
C          ND........ROW DIMENSION OF THE VARIABLE FROM THE
C                    DIMENSION STATEMENT IN THE CALLING PROGRAM.
C          REAL......LOGICAL CONSTANT EQUAL TO TRUE IF THE VARIABLE
C                    MODE IS REAL AND EQUAL TO FALSE IF THE
C                    VARIABLE MODE IS INTEGER.
C
C               NON DIMENSIONED VARIABLES ARE ENTERED WITH N, M,
C               AND ND EQUAL TO 1.  SINGLE DIMENSIONED VARIABLES
C               HAVE THE NORMALLY UNUSED DIMENSION SET TO 1.
C
C
C     COMMON----------LABELED CONTROL
C          DSP.......THE INTEGER TAPE NUMBER OR A LEFT JUSTIFIED,
C                    ZERO FILLED HOLLERITH CONSTANT INDICATING WHICH
C                    FILE THE SUBROUTINE IS TO WRITE OUT ON.  THIS
C                    FILE SHOULD BE A CONNECTED FILE.
C          BATCH.....A LOGICAL CONSTANT WITH A VALUE OF
C                    TRUE IF THIS ROUTINE IS BEING USED IN A BATCH
C                    JOB AND A VALUE OF FALSE IF USED FOR AN
C                    INTERACTIVE JOB.  (IF ANY FORMAT ERROR IS
C                    DETECTED WHEN RUNNING A BATCH JOB THE
C                    DIAGONISTIC IS PRINTED AND THEN AN ABORT
C                    TERMINATION IS EXECUTED)
C
C     TAPE5 IS THE NORMAL INPUT FILE.  IT SHOULD BE CONNECTED
C     AND/OR EQUIVALENCED TO INPUT.
C***********************************************************************
```

```
      SUBROUTINE READCOM(COM,NUM)
C***********************************************************************
C          THE PURPOSE OF THIS SUBROUTINE IS TO READ IN AND VALIDATE
C     ALPHANUMERIC AND SYMBOLIC COMMANDS OF 9 CHARACTERS OR LESS.
C     COMMANDS MAY HAVE SINGLE IMBEDED BLANKS, AND LEADING BLANKS
C     ARE IGNORED.  COMMANDS ARE SEPARATED BY COMMAS,SLASHES, OR TWO
C     OR MORE BLANKS.  COMMANDS PRECEEDED BY A COMMA ARE CONSIDERED
C     TO BE PARAMETERS OF THE LAST PREVIOUS COMMAND NOT PRECEEDED BY
C     A COMMA.  THIS ROUTINE IS MENT FOR SINGLE COMMAND STRING ENTRIES,
C     BUT MULTIPLE COMMANDS CAN BE ENTERED (SEPARATED BY SLASHES OR
C     2 OR MORE BLANKS) AS LONG AS THE TOTAL CHARACTER COUNT DOES
C     NOT EXCEED 80.  A $ ENTERED AT ANY TIME WILL CAUSE AN ABORT
C     CONDITION AS EXPLAINED BELOW.
C          COMMANDS ARE VALIDATED AGAINST A HOLLERITH ARRAY (LIST) OF
C     LEGAL COMMANDS.  ABBREVIATIONS ARE ALLOWED AS LONG AS THEY
C     ARE UNIQUE.  COMMAND STRING FORMAT IS NOT CHECKED.
C
C
C          THE CALL PARAMETERS AND THE LABELED COMMON BLOCKS REQUIRED
C     ARE EXPLAINED BELOW.
C
C     PARAMENTERS----------
C          COM..AN ARRAY DIMENSIONED AT LEAST NUM.  ON RETURN THIS
C               ARRAY WILL CONTAIN INDICES FOR THE LIST ARRAY
C               CORRESPONDING TO THE COMMANDS ENTERED.  A NEGATIVE
C               VALUE INDICATES THAT THIS COMMAND IS A PARAMENTER
C               IN A COMMAND STRING.
C          NUM..AN INTEGER INDICATING ON INPUT THE MAXIMUM NUMBER OF
C               COMMANDS TO BE READ, AND ON OUTPUT THE NUMBER OF COMMANDS
C               ACTUALLY READ.
C
C                    A ZERO WILL BE RETURNED IN BOTH COM(1) AND NUM
C                    IN CASE OF AN ABORT.
C
```

```
C
C    COMMON-----------LABELED CONTROL
C          DSP.......THE INTEGER TAPE NUMBER OR A LEFT JUSTIFIED,
C                    ZERO FILLED HOLLERITH CONSTANT INDICATING WHICH
C                    FILE THE SUBROUTINE IS TO WRITE OUT ON.  THIS
C                    FILE SHOULD BE A CONNECTED FILE.
C          BATCH.....A LOGICAL CONSTANT WITH A VALUE OF
C                    TRUE IF THIS ROUTINE IS BEING USED IN A BATCH
C                    JOB AND A VALUE OF FALSE IF USED FOR AN
C                    INTERACTIVE JOB.  (IF ANY VALIDITY ERROR IS
C                    DETECTED WHEN RUNNING A BATCH JOB THE
C                    DIAGONISTIC IS PRINTED AND THEN COM(1) AND
C                    NUM ARE SET TO ZERO AND A RETURN IS EXECUTED.
C
C
C    COMMON-----------LABELED LISTING
C          NL........THE NUMBER OF ACTIVE ELEMENTS IN THE ARRAY
C                    LIST.
C          LIST......AN ARRAY DIMENSIONED AT LEAST NL CONTAINING THE
C                    COMMAND NAMES TO BE CHECKED.  EACH LIST(K) IS A
C                    HOLLERITH CONSTANT WITH THE 'NAME' OF THE COMMAND IN
C                    THE FIRST 9 CHARACTERS, LEFT JUSTIFIED, AND THE
C                    NUMBER OF LETTERS IN THE 'NAME' IN THE LAST
C                    CHARACTER.
C                    EXAMPLE....
C                    A COMMAND  10HTIME     D
C                         THE DISPLAY CODE FOR D IS 04, WHICH WHEN
C                         THE REST OF THE WORD IS MASKED
C                         IS THE INTEGER 4 - THE NUMBER OF
C                         LETTERS IN TIME.
C***********************************************************************
```

```
      SUBROUTINE PRINTR(NA,NAM,VAR,N,M,ND)
C*********************************************************************
C     \   THE PURPOSE OF THIS SUBROUTINE IS TO PRINT THE VALUES OF A NON,
C    ONE, OR TWO DIMENSIONAL VARIABLE.  THE VARIABLE MAY BE PRINTED TO
C    TAPE 6 OR TO A FILE DESIGNATED BY THE USER OR TO BOTH OR TO NEITHER.
C    THE DECISION ON WHERE TO PRINT IS BASED ON A CODE PASSED IN THE
C    COMMON BLOCK LISTOUT FOR EACH VARIABLE. ONCE THE DECISION IS MADE
C    AS TO WHERE TO PRINT, THE SUBROUTINE LISTER IS CALLED TO DO THE
C    ACTUAL PRINTING.  THIS SUBROUTINE IS CALLED WITH THE ENTRY
C    POINT PRINTR FOR REAL VARIABLES AND THE ENTRY POINT PRINTI
C    FOR INTEGER VARIABLES
C
C
C         THE CALL PARAMETERS AND LABELED COMMON BLOCKS REQUIRED ARE
C    EXPLAINED BELOW.
C
C    PARAMETERS----------
C         NA...THE INDEX FOR THE ARRAY LISTO CORRESPONDING TO THE
C              VARIABLE TO BE PRINTED.
C         NAM..HOLLERITH CONSTANT WITH THE NAME OF THE VARIABLE IN THE
C              FIRST 9 CHARACTERS, LEFT JUSTIFIED, AND THE NUMBER OF
C              LETTERS IN THE NAME IN THE LAST CHARACTER; THIS LAST
C              CHARACTER WILL BE AN INTEGER VALUE FOR THE NUMBER OF
C              LETTERS IN THE NAME WHEN THE PRECEEDING NINE CHARACTERS
C              ARE MASKED.  IF NO NAME IS TO BE PRINTED WITH THE VARIABLE
C              THIS SHOULD BE 10H         A.
C         VAR..THE VARIABLE TO BE PRINTED
C         N....THE ROW DIMENSION OF THE VARIABLE TO BE PRINTED
C         M....THE COLUMN DIMENSION OF THE VARIABLE TO BE PRINTED
C         ND...THE ROW DIMENSION OF THE VARIABLE FROM THE DIMENSION
C              STATEMENT IN THE CALLING PROGRAM.
C
C    COMMON----------LABELED LISTOUT
C         NL........THE NUMBER OF ACTIVE ELEMENTS IN THE ARRAY LISTO
```

```
C        LISTO.....AN ARRAY DIMENSIONED AT LEAST NL WITH AN ELEMENT
C                  FOR EACH VARIABLE TO BE PRINTED.  IF THE VALUE
C                  IS A 1 THE VARIABLE WILL NOT BE PRINTED.  IF THE
C                  VALUE IS A 2 THE VARIABLE WILL BE PRINTED TO THE
C                  FILE SPECIFIED BY DSP IN LABELED COMMON -CONTROL-.
C                  IF THE VALUE IS A 3 THE VARIABLE WILL BE PRINTED
C                  TO BOTH TAPE 6 AND THE FILE DSP.
C                  IF THE VALUE IS A 4 THE VARIABLE WILL BE PRINTED
C                  TO TAPE 6 ONLY.
C
C    COMMON----------LABELED CONTROL
C        DSP..AN INTEGER SPECIFING A TAPE NUMBER FOR THE USERS DESIRED
C             PRINT FILE OR A LEFT JUSTIFIED, ZERO FILLED NAME OF THE
C             USERS FILE. (USUALLY TAPE 5 IS DISCONNECTED FOR HARD
C             COPY STORAGE OF THE PRINTED VALUES AND THE FILE DSP IS
C             CONNECTED TO THE TERMINAL FOR IMMEDIATE DISPLAY TO THE
C             USER)
C
C    COMMON----------LABELED TITLE
C        ITITLE...DIMENSIONED 5, THIS PROVIDES SPACE TO PRINT A 50
C                 CHARACTER TITLE ABOVE THE VARIABLE PRINTOUT.  IF
C                 THE ARRAY ITITLE IS NON BLANK THE NAME OF THE
C                 VARIABLE WILL NOT BE PRINTED.  THE ARRAY IS SET
C                 TO ALL BLANKS AFTER EACH CALL TO THIS SUBROUTINE.
C***********************************************************************
```

```
      SUBROUTINE LISTER(NAME,VAR,N,M,ND,FOR,DST)
C***********************************************************************
C           THIS SUBROUTINE IS CALLED BY THE SUBROUTINE PRINTR TO TO THE
C     ACTUAL PRINTING OF THE VARIABLES.  THE PARAMETERS USED IN THE
C     CALL ARE GENERATED BY PRINTR AND WILL NOT BE EXPLAINED HERE, AS
C     THE TWO SUBROUTINES ARE MENT TO BE USED TOGETHER.
C           THIS SUBROUTINE USES ONE OTHER LABELED COMMON BLOCK TO
C     DETERMINE THE DESIRED FORMAT FOR THE VARIABLE TO BE PRINTED
C     IN. THESE TERMS ARE EXPLAINED BELOW.
C
C
C     COMMON----------LABELED FORMS
C           NDREP.....AN INTEGER INDICATING THE NUMBER OF COLUMNS TO BE
C                     PRINTED ON ONE LINE ON THE FILE DESIGNATED BY THE
C                     USER WITH DSP.
C           NDNSD.....AN INTEGER INDICATING THE NUMBER OF SIGNIFICANT
C                     DIGITS TO BE PRINTED TO THE FILE DSP.
C           NDFW......AN INTEGER INDICATING THE FIELD WITH OF THE FORMAT
C                     USED FOR PRINTING THE VARIABLE TO THE FILE DSP.
C                     THIS NUMBER MUST BE 7 MORE THAN NDNSD.
C           NOREP.....AN INTEGER SIMILAR TO NDREP, EXCEPT THIS ONE
C                     CORRESPONDS TO THE FILE TAPE 6.
C           NONSD.....AN INTEGER SERVING THE SAME FUNCTION AS NDNSD, ONLY
C                     FOR TAPE 6
C           NOFW......AN INTEGER SPECIFING THE FILED WIDTH FOR TAPE 6 PRINT
C***********************************************************************
```

```
MATRIX MANIPULATION SUBROUTINE CALL PARAMETERS

CALL ADDMAT(A,NAD,N,M,B,NBD,C,NCD)      **  A = B + C  **
        A -- OUTPUT MATRIX
        NAD -- ROW DIMENSION OF A FROM DIMENSION STATEMENT IN
                CALLING PROGRAM
        N -- ROW DIMENSION OF A,B, AND C
        M -- COLUMNS DIMENSION OF A,B, AND C
        B & C -- MATRICES TO BE ADDED
        NBD -- ROW DIMENSION OF B FROM DIMENSION STATEMENT IN
                CALLING PROGRAM
        NCD -- ROW DIMENSION OF C FROM DIMENSION STATEMENT IN
                CALLING PROGRAM

CALL SUBMAT(A,NAD,N,M,NB,NBD,C,NCD)      **  A = B - C  **
        SAME PARAMETERS AS FOR ADDMAT EXCEPT THAT NOW MATRIX
        C IS BEING SUBTRACTED FROM MATRIX B

CALL TRANPOS(A,NAD,N,M,B,NBD)      **  A = B(TRANSPOSE)
        A -- OUTPUT MATRIX
        NAD -- ROW DIMENSION OF A FROM DIMENSION STATEMENT IN
                CALLING PROGRAM
        N -- ROW DIMENSION OF A AND COLUMN DIMENSION OF B
        M -- COLUMN DIMENSION OF A AND ROW DIMENSION OF B
        B -- INPUT MATRIX
        NBD -- ROW DIMENSION OF B FROM DIMENSION STATEMENT IN
                CALLING PROGRAM

CALL COPYAB(A,NAD,N,M,B,NBD)      **  A = B  **
        A, NAD, B, AND NBD ARE THE SAME AS FOR TRANPOS
        N -- ROW DIMENSION OF A AND B
        M -- COLUMN DIMENSION OF A AND B

CALL FPRESET(A,NAD,N,M,PRE)      **  A = PRE  **
```

```
        A -- OUTPUT MATRIX
        NAD -- ROW DIMENSION OF A FROM DIMENSION STATEMENT IN
               CALLING PROGRAM
        N -- ROW DIMENSION OF A
        M -- COLUMN DIMENSION OF A
        PRE -- REAL VALUE TO BE SET IN EACH ELEMENT OF A

CALL IDENT(A,NAD,N)      **  A = I  **
        A -- OUTPUT MATRIX
        NAD -- ROW DIMENSION OF A FROM DIMENSION STATEMENT IN
               CALLING PROGRAM
        N -- ROW AND COLUMN DIMENSION OF A

CALL MAT2MPY(A,NAD,N,M,B,NBD,N1,C,NCD)      $$  A = B * C  $$
        A -- OUTPUT MATRIX
        NAD -- ROW DIMENSION OF A FROM DIMENSION STATEMENT IN
               CALLING PROGRAM;;N -- ROW DIMENSION OF A AND B
        M -- COLUMN DIMENSION OF A AND C
        B & C -- MATRICES TO BE MULTIPLIED
        NBD -- ROW DIMENSION OF B FROM DIMENSION STATEMENT IN
               CALLING PROGRAM
        N1 -- COLUMN DIMENSION OF B AND ROW DIMENSION OF C
        NCD -- ROW DIMENSION OF C FROM DIMENSION STATEMENT IN
               CALLING PROGRAM

CALL MAT3MPY(A,NAD,N,M,B,NBD,N1,C,NCD,N2,D,NDD)
                          $$  A = B * C * D  $
        A, NAD, N, NBD, N1, AND NCD ARE THE SAME AS FOR MAT2MPY
        M -- COLUMN DIMENSION OF A AND D
        B,C, & D -- MATRICES TO BE MULTIPLIED
        N2 -- COLUMN DIMENSION OF C AND ROW DIMENSION OF D
        NDD -- ROW DIMENSION OF D FROM DIMENSION STATEMENT IN
               CALLING PROGRAM
```

```
CALL INVFRT(A,NAD,B,NBD,N,IFAIL)      **  A = B(INVERSE)
        A -- OUTPUT MATRIX
        NAD -- ROW DIMENSION OF A FROM DIMENSION STATEMENT IN
                 CALLING PROGRAM
        B -- INPUT MATRIX  (THIS MATRIX DESTROYED DURING
               EXECUTION)
        NBD -- ROW DIMENSION OF B FROM DIMENSION STATEMENT IN
                 CALLING PROGRAM
        N -- ROW AND COLUMN DIMENSIONS OF A AND B
        IFAIL -- LOGICAL VARIABLE RETURNED TRUE IF MATRIX IS
                 SINGULAR
                 (1)   SUBROUTINE IDENT IS CALLED BY THIS ROUTINE
                 (2)   ONE ELEMENT OF LABELED COMMON -CONTROL- IS
                       USED TO INDICATE WHICH FILE THE ERROR
                       MESSAGE SHOULD BE WRITTEN TO IF THE MATRIX
                       IS SINGULAR
```

Appendix D

Card Sequence Required To Utilize FORTRAC

The use of the batch mode program FORTRAC is described in the preliminary report "FORTRAC: A Software Package for the Design of Multivariable Digital Control Systems" which is available from the FGL branch of the Flight Dynamics Lab. The one point that needs to be expanded on is the card sequence required for the different types of problems that can be run with the program. The following pages give the card sequence required for the three types of problems; the regulator, tracker, and disturbance rejector.

The following limits are in effect with the presently dimensioned master program.

(1) The maximum number of states including augmenting integrators allowed is 7.

(2) The maximum number of controls is 4.

(3) The maximum number of augmenting integrators allowed is 3.

(4) The maximum number of samples that can be displayed in either the discrete or continuous simulation is 33.

These limits can be increased by increasing the dimensions on the 133 variables in the master program.

Card Sequence for the Regulator

1 N,M,NCC,NDD,NYY,NPP

N	= number of states
M	= number of controls
NCC	= 0
NDD	= 0
NYY	= number of outputs
NPP	= 0

The elements of the state matrix, one row per card

The elements of the control matrix, one row per card

The elements of the output matrix, one row per card

T The sampling interval to be used

The desired closed-loop eigenvalues, N cards with a real and an imaginary part on each card

---

0 No observer will be designed

K An integer indicating the power to which the closed-loop matrix should be raised

---

NDS Number of time intervals required in the discrete-time simulation

---

N initial conditions for the states, all on one card

EPS,NDS,NSIDS

EPS	= the accuracy to be used in the Kutta-Merson subroutine (suggest approximately .0000001)
NDS	= number of sample-time intervals required in the continuous-time simulation
NSIDS	= number of divisions required in each sample-time interval

---

N initial conditions for the states, all on one card

Card Sequence for the Tracker

1 N,M,NCC,NDD,NYY,NPP

N = number of states
M = number of controls
NCC = number of commands
NDD = 0
NYY = number of outputs
NPP = number of outputs integrated (equal to NCC)

The elements of the state matrix, one row per card

The elements of the control matrix, one row per card

The elements of the command matrix, one row per card
(this must be entered even if the matrix is null)

The elements of the output matrix, one row per card

The elements of the matrix defining the outputs that will track the input command

---

T The sampling interval to be used

The number of integrators desired for each integrated output, all entered on one card

---

The desired closed-loop eigenvalues, (N + the sum of the numbers entered on the preceeding card) cards with a real and an imaginary part on each card

---

0 No observer will be desinged

K An integer indicating the power to which the closed-loop matrix should be raised

---

NDS Number of time intervals required in the discrete-time simulation

---

N initial conditions for the states, all on one card

Initial conditions for the integrators, all on one card

EPS,NDS,NSIDS

Same as for the regulator

N initial conditions for the states, all on one card

Initial conditions for the integrators, all on one card

Card Sequence for the Disturbance Rejector

1 N,M,NCC,NDD,NYY,NPP

N =	number of states
M =	number of controls
NCC =	0
NDD =	number of disturbances
NYY =	number of outputs
NPP =	number of outputs integrated (number of outputs to reject disturbances)

The elements of the state matrix, one row per card

The elements of the control matrix, one row per card

The elements of the disturbance matrix, one row per card

The elements of the output matrix, one row per card

The elements of the matrix defining the outputs that will reject disturbances

T The sampling interval to be used

All remaining cards are the same as for the tracker

Vita

James A. Colgate was born on 13 February 1948 in Fort Collins, Colorado. He moved to Miami, Arizona at the age of 7 and graduated from Miami High School in 1966. He graduated from the United States Air Force Academy in 1970 with a Bachelor of Science Degree in Electrical Engineering. After graduation, he entered pilot training at Williams AFB, Arizona and completed training in July 1971. From August 1971 to September 1974 he served as a WC-135 pilot in the 55th Weather Reconnaissance Squadron. He attended Squadron Officer School in residence from September to December 1974 and then spent the next year at a remote radar site, Indian Moutain Alaska, as a Weapons Controller. He entered the School of Engineering, Air Force Institute of Technology, in June 1976.

Permanent address: 110 Miami Gardens
Miami, Arizona 85539

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 14 AFIT/GGC/EE/77D-1	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 INTERAC - AN INTERACTIVE SOFTWARE PACKAGE FOR DIRECT DIGITAL CONTROL DESIGN		5. TYPE OF REPORT & PERIOD COVERED 9 Master's thesis, [redacted]
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 James A./Colgate [redacted]		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE 11 Dec [redacted] 77
		13. NUMBER OF PAGES 12 [redacted] 123 P.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release; IAW AFR 190-17

[illegible]

JERRAL F. GUESS, Captain, USAF
Director of Information

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Control
Software
Discrete Control
Interactive Programming

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The purpose of this investigation is to develop an interactive user oriented software package for direct digital control system design. The batch mode program, FORTRAC, developed by Professor Brian Porter of the University of Salford, England, is the source of the computational algorithms used. This report details the input, output, and program sequence control software developed to produce an efficient, user oriented interactive package. → over

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified hC
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

φ12225

The package is very forgiving of user errors and gives the user complete control over what data to input, what data will be output, and which parts of the program are executed. The package is quite useful in the design process for discrete-time time-optimal control systems, where many possible control parameter variations must be examined.